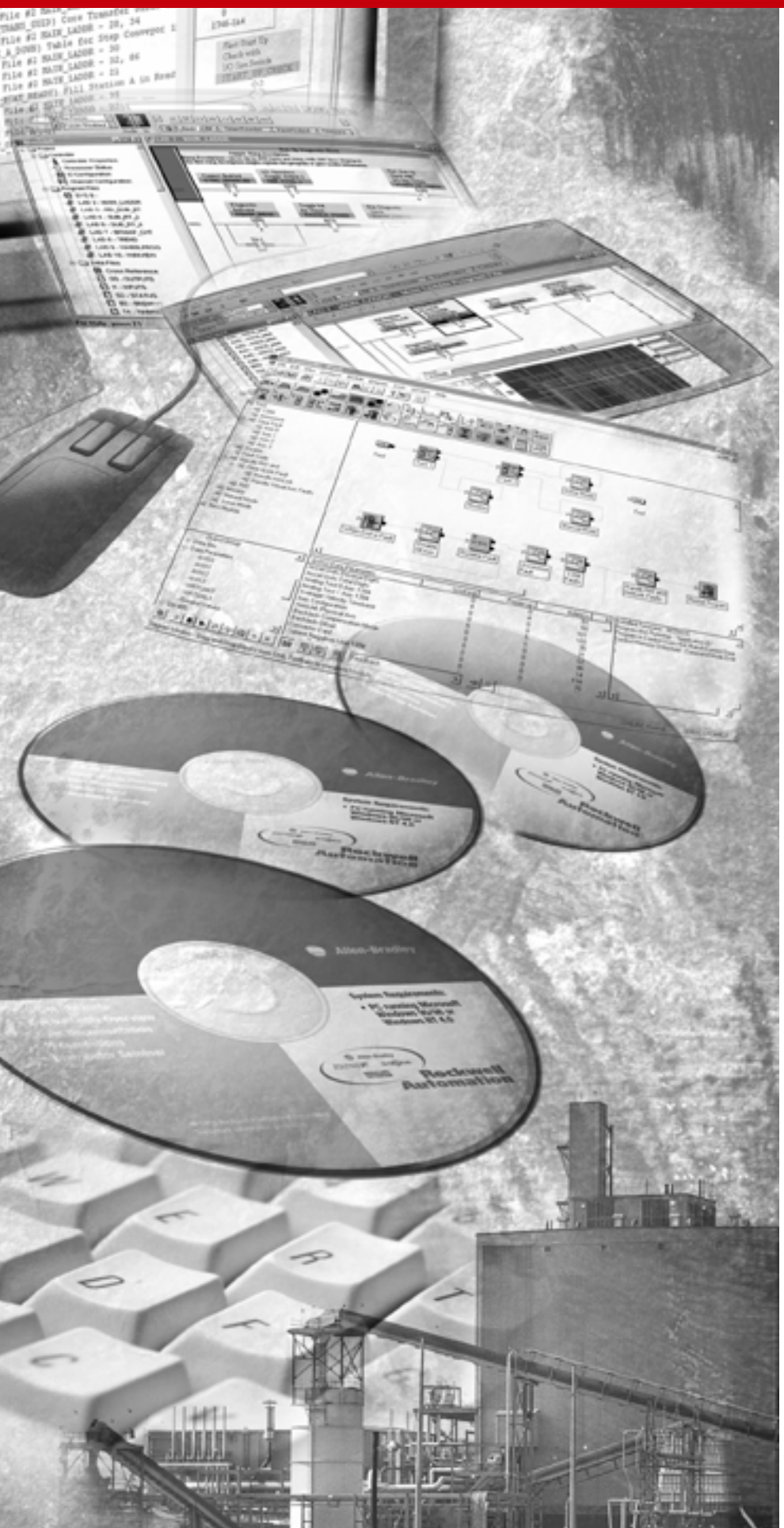


Allen-Bradley

ProcessLinx OPC Server to Server Software

1757-PROCESSLINX

User Manual

Rockwell Automation

Important User Information

Because of the variety of uses for the products described in this publication, those responsible for the application and use of this control equipment must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards.

The illustrations, charts, sample programs and layout examples shown in this guide are intended solely for purposes of example. Since there are many variables and requirements associated with any particular installation, Allen-Bradley does not assume responsibility or liability (to include intellectual property liability) for actual use based upon the examples shown in this publication.

Allen-Bradley publication SGI-1.1, *Safety Guidelines for the Application, Installation and Maintenance of Solid-State Control* (available from your local Allen-Bradley office), describes some important differences between solid-state equipment and electromechanical devices that should be taken into consideration when applying products such as those described in this publication.

Reproduction of the contents of this copyrighted publication, in whole or part, without written permission of Rockwell Automation, is prohibited.

Throughout this manual we use notes to make you aware of safety considerations:

ATTENTION

Identifies information about practices or circumstances that can lead to personal injury or death, property damage or economic loss

Attention statements help you to:

- identify a hazard
- avoid a hazard
- recognize the consequences

IMPORTANT

Identifies information that is critical for successful application and understanding of the product.

Allen-Bradley is a trademark of Rockwell Automation, Inc.

Introduction

This document provides an overview of the ProcessLinx application. ProcessLinx allows two OPC servers to communicate. Normally a server talks to a client and visa versa. A server cannot talk to another server because the interfaces are incompatible. ProcessLinx is specilized software that allows two servers to communicate.

Overview

The OPC (OLE for Process Control) Interface within ProcessLinx acts as an OPC Client to a ProcessLogix OPC Server as well as to another OPC server. To interface from ProcessLogix to another server, a ProcessLinx TagMap configuration must be created. This configuration maps the items from one system to the other. This is accomplished through the exchange of data with entities known as tags or OPC items. The OPC items can be logically Grouped together. ProcessLinx supports both local and remote OPC Servers.

This interface supports bi-directional data flow and supervisory control between ProcessLogix and other control systems. It enables operators to control digital and analog control loops implemented within ProcessLogix controllers from external control system operator consoles. Likewise, ProcessLogix operator stations may also be used for operator control of digital and analog control loops implemented in other control system processors. These capabilities depend on the functionality supported by the external OPC Server being utilized. ProcessLinx is fully compliant with Versions 2.0 and 1.0a OPC Data Access Servers.

Functional Description

The ProcessLinx interface runs under the Windows NT/Windows 2000 Professional operating system(s). Two modes of operation are associated with the interface, which are “configure” and “runtime”. In configure mode a user can change settings and create TagMaps. In runtime mode the interface uses the currently loaded configuration to communicate with the external OPC Server(s). ProcessLinx supports the following:

- OPC Server configuration dialog allows user to define the external OPC Server by browsing and selecting an existing OPC Server.
- Supports redundant ProcessLogix OPC Servers as well as redundant external OPC servers by recognizing server fail-over mechanisms.
- Simple TagMap definition dialog allows mapping of an external OPC Server tag to a ProcessLogix OPC Server tag. Tag Map definition will include attributes such as OPC Item names, description and data source. Optionally, scaling conversion using translation tables can be defined to translate values from one system to the other.
- The number of user-defined TagMap definitions is unlimited.

- Tag Map configurations are stored in a serialized compact binary file.
- Supports ability to easily import and export ProcessLinx Tag Map configurations to/from comma delimited (*.csv) ASCII files, which can be created and/or edited with Microsoft Excel, Microsoft Access or any text editor.
- Error and informational messages displayed in the main window in runtime along with optional user-configurable specification maintaining this data in daily log files.
- Runtime tag monitoring ability to view and change (read/write) data values in either OPC Server from main client window.

Rockwell Automation Technical Support

Rockwell Automation offers support services worldwide, with over 75 sales/support offices, 512 authorized distributors, and 260 authorized systems integrators located throughout the United States alone, plus Rockwell Automation representatives in every major country in the world.

Local Product Support

Contact your local Rockwell Automation representative for:

- sales and order support
- product technical training
- warranty support
- support service agreements

Technical Product Assistance

If you need to contact Rockwell Automation for technical assistance, first call your local Rockwell Automation representative, then:

- Network Pre-sales Hotline, 440.646.3638 (3NET)
- Post-sales Technical Support, 440.646.5800
- Web Links http://www.ab.com — as a registered member, open to http://www.ab.com/mem/technotes/techmain.html

Your Questions or Comments about This Manual

If you find a problem or have a comment about this manual, please notify us of it on the enclosed Publication Problem Report (at the back of this manual).

If you have any suggestions about how we can make this manual more useful to you, please contact us at the following address:

Rockwell Automation, Allen-Bradley Company, Inc.
Control and Information Group
Technical Communication
1 Allen-Bradley Drive
Mayfield Heights, OH 44124-6118

Conventions

Terms and Type Representations

The following table summarizes the terms and type representation conventions used in this Guide.

Term/Type Representation	Meaning	Example
click, click on, click [**button name**]	Click left mouse button once. (Assumes cursor is positioned on object or selection.)	Click **Apply**.
double-click	Click left mouse button twice in quick succession. (Assumes cursor is positioned on object or selection.)	Double-click the Pipe.
drag	Press and hold left mouse button while dragging cursor to new screen location and then release the button. (Assumes cursor is positioned on object or selection to be moved.)	Drag the file icon onto the new folder window.
right-click	Click right mouse button once. (Assumes cursor is positioned on object or selection.)	Right-click, and the pop-up menu appears.
select	click to highlight a menu item or list choice, or click on a button.	Select the OPC Item from the list box.
<F1>	Keys to be pressed are shown in angle brackets.	Press <F1> to view the online Help.
<Ctrl>+<C>	Keys to be pressed together are shown with a plus sign.	Press <Ctrl>+<C> to configure the Pipe.
File ⇒ **New**	Shows menu selection as menu name followed by menu selection.	Click **Edit** ⇒ **Add** New Group to create a Group.
>D:\setup.exe<	Data to be keyed in at prompt or in an entry field.	Key in this path location >D:\setup.exe<.

Table of Contents

Preface

Introduction P-1
Overview. P-1
Functional Description P-1
Rockwell Automation Technical Support P-3
Local Product Support P-3
Technical Product Assistance P-3
Your Questions or Comments about This Manual P-3
Conventions P-4
Terms and Type Representations P-4

Chapter1
Setup Procedures

Installation 1-1
Configuring DCS Components. 1-1
Installing OPC Servers 1-1
Defining an Interface Control Strategy. 1-2
Creating OPC Server Configurations 1-2
Configure the OPC Server(s). 1-3
Installing ProcessLinx 1-3
Preparing the Client's User Accounts 1-3
Running Updates on the ProcessLinx Client's Registry. 1-4

Chapter 2
Getting Started

ProcessLinx Version 2-1
Configure Mode 2-1
Runtime Mode 2-2
Configure ProcessLinx Options 2-3

Chapter 3
ProcessLinx TagMap Configuration

Creating a ProcessLinx Configuration 3-2
Opening a ProcessLinx Configuration 3-2
Creating a Pipe 3-2
Creating a Group 3-8
Creating a TagMap 3-11
Creating a ProcessLinx TagMap Configuration 3-15
Data Conversion/Scaling. 3-17
OPC Items 3-17
Translation Table 3-21
Editing a Pipe, Group or TagMap. 3-22
Copying/Moving TagMaps between Groups. 3-22
Multiple TagMap Configuration 3-23
Testing a ProcessLinx Configuration 3-24

Chapter 4
ProcessLinx Runtime Operation

Monitoring Values 4-1
Reading OPC Values 4-1
Writing OPC Values. 4-2
Redundant OPC Servers / Fail over 4-3
Server Status 4-4
Verifying OPC Tags 4-5
Ping 4-5
OPCShutdown 4-5

Chapter 5
Import/Export

CSV File Formats 5-3
Pipe Declarations 5-4
Group Declarations 5-5
TagMap Declarations. 5-6
Translation Table declarations 5-7

Chapter 6
Error Diagnosis and Debugging

Quick Setup 6-1
Debug Operations 6-3

Chapter 7
Using DCOM OPC Servers

OPC Servers on Windows NT/Windows 2000 7-1
OPC Servers on Windows98 / Windows 95 7-2

Setup Procedures

Installation

This chapter contains an overview of the setup procedure and the steps you need to take to set up a configuration prior to using ProcessLinx.

Configuring DCS Components

IMPORTANT You are required to configure the DCS components before installing ProcessLinx. If the DCS components are not configured correctly, server communications will not succeed.

All DCS Components must be configured first in the setup procedure. This includes installation and setup of the following:

- **ProcessLogix System** – The ProcessLogix system must be setup and installed on the PC used to access the ProcessLogix OPC Server.
- **DCS** – The other DCS system is assumed to be setup prior to performing the steps in the section Creating a ProcessLinx TagMap Configuration on page 3-10.

Installing OPC Servers

Install the desired OPC servers. If a remote DCOM connection is required, verify the PC is setup on the network with the TCP/IP protocol installed. The OPC Servers are specified below:

ProcessLogix OPC Server

All of the software components of the ProcessLogix OPC Server are installed during the ProcessLogix Server installation.

IMPORTANT The OPC Server option MUST be included with your ProcessLogix license.

In order for ProcessLinx to connect to the ProcessLogix OPC Server over the network, the *OPC Server Connect* option must be installed on the client PC.

Refer to the ProcessLogix Knowledge Builder: Navigate in Knowledge Builder to ⇒*Server and Client Configuration Guide* ⇒*Using ProcessLogix Data in other Applications* ⇒ *ProcessLogix OPC Support.*

External OPC Server

Consult the External OPC Server documentation for correct installation and setup on the desired node. This does not have to be the same PC or node running ProcessLinx. Verify the External OPC Server operation separately.

Defining an Interface Control Strategy

Define the site specific DCS Control Strategy. This step assumes that you have two working DCS configurations, one in ProcessLogix, and the other in an external DCS system. This step includes defining exactly which points (control loops, pumps, motors, etc.) are to be controlled and interfaced and by which system. This step also tentatively outlines the direction and flow of data and/or specifies which system is in control of each interfaced point.

It is emphasized that if this step is not performed, data flow direction and determination of what point is coming from where may get very confusing in the future. A table laying out all desired interface points, their names within both systems, and data flow directions should be created and documented for future reference.

Creating OPC Server Configurations

Before a ProcessLinx configuration can be realized, an OPC Server database must be built which defines references to I/O points in the underlying desired hardware or to control points. With ProcessLogix, this step is not required since control points defined using Control Builder are immediately available from the ProcessLogix OPC Server. No OPC Server configuration must be built for those points in the ProcessLogix OPC Server. This includes points brought in and defined from other interfaces using the Quick Builder program.

The External OPC Server configuration (database) must be configured to define exactly what pieces of hardware or blocks you are interested in. Consult the External OPC Server documentation for setup.

Configure the OPC Server(s)

Before ProcessLinx can work properly in runtime, each OPC Server must be installed and configured correctly. Follow the installation procedure and setup specified with the documentation for the external OPC Server. A TCP/IP connection must also exist between the NT ProcessLogix node and any remote node running an external OPC server for which you wish ProcessLinx to connect with. Refer to Using DCOM OPC Servers on page 7-1 to setup and configure remote OPC servers.

Installing ProcessLinx

Do the following to install the ProcessLinx software:

1. Insert the ProcessLinx CD.
2. Select **Start ⇒ Run**.
3. Enter D(CDROM drive letter):setup.exe.
4. Accept the License Agreement.
5. Follow the rest of the dialog boxes to install ProcessLinx in the default locations.
6. To complete the installation, insert the floppy disk with the License Activation.
7. Follow the instructions for moving the License Activation to the PC.

   ProcessLinx installation is complete.

Preparing the Client's User Accounts

The easiest way to avoid DCOM security problems is to have the same user account and password on all machines, DCOM security is bypassed. It is possible however, to allow a remote OPC Client to access the PLX OPC Server when the correct permissions are specified. This requires user and group privileges and DCOMCFG settings to be set appropriately. This is not a trivial undertaking and most people (including network administrators) have no idea how to do it properly.

Most of the time OPC is set up with user accounts on the Client machine and Server machine having the same name and password. This is not specific to ProcessLinx. It is a Microsoft DCOM issue. OPC is written using COM so it has inherited all the DCOM flaws.

ProcessLinx is not dependent on a specific version of ProcessLogix. It works the same in R320 as it does in R500. Refer to your Microsoft documentation on how to add user accounts.

Both the ProcessLogix Client and the Server must have both the MNGR and

ps_user accounts on them. Refer to the sections in Knowledge Builder, *About ProcessLogix-To-OPC Server/Client Communications, Communicating with 3rd Party Clients*, and *Creating a MNGR Account for a Third-Party Client*.

IMPORTANT The following instructions apply only to Windows 2000 and Windows XP. Windows NT is not supported.

For example, if the ps_user account is not the client, you must add it by doing the following:

1. Right click on **My Computer**.
2. Select **Manage ⇒ Local Users and Groups ⇒ Users**.
3. Right click on the **Users** folder.
4. Select **New User**.
5. Add a PS_USER account.
6. On the General Tab, check **Password Never Expires**.
7. Click the Member Of Tab and be sure Administrators and Users are visible.

   The PS_USER account is now setup on the client.

Running Updates on the ProcessLinx Client's Registry

Because OPC is COM based, the Client needs the GUID (Globally Unique IDentifier) of the OPC Server. If the Client and Server reside on the same box, the registry already contains the Server's GUID so the Client can easily find it. If the Server resides on a remote computer, the Client's registry does not contain the Server's GUID.

If you have ProcessLinx installed on a third machine you will need to make sure you have the GUIDs of both OPC servers located in the ProcessLinx machine. You will need to follow similar steps as noted below. The following steps put the Server's GUID into the Client's registry.

Setting up communication between the OPC Client and OPC Server

Follow these instructions to setup communications between the OPC Client and the OPC Server:

1. Select **Start ⇒ Run**.

2. Type regsvr32 opcproxy.dll in the Run dialog box.

IMPORTANT Unless opcproxy.dll is in the Windows\System32 directory the above command will fail because it can't find the OPCPROXY.DLL. You need to specify the full path the the DLL (i.e. regsvr32 "C:\Program Files\Common Files\OPC Foundation\opcproxy.dll")

3. On the server machine, go to the C:\Honeywell\Client\Opcserverconnect, folder and copy the HSCOPCCONNECT.REG & HSCOPCSERV.REG files to a floppy diskette.

4. On the ProcessLinx Client machine, create a directory called C:\Honeywell\Client.

5. Copy the hscopcconnect.reg file and the hscopcServ.reg file from the floppy, to the C:\Honeywell\Client directory of the ProcessLinx Client.

6. Using Notepad, open then edit the hscopcconnect.reg file.

7. Edit line#19 section. Change the entry
   from: @="d:\\hwiac\\programs\\tps\\hci\\hscopcserv.exe

   to: @="c:\honeywell\client\hscopcserv.exe

8. Go into the C:\Honeywell\Client directory on the Client.

9. Run (double-clicking) each of the .REG files in Step 3, HSCOPCCONNECT.REG & HSCOPCSERV.REG, one at a time.

   All registry changes that tell the Client, the Servers GUID have now been made to the registry.

Notes:

Getting Started

This section provides the required information to setup and configure both the ProcessLinx software as well as the communications interface to the ProcessLogix OPC Server and the external OPC Server.

ProcessLinx Version

The version of the ProcessLinx can be found by selecting the Help menu in the main window. The version label below displays Version 2 and Build # is 68, built on January 10, 2003.

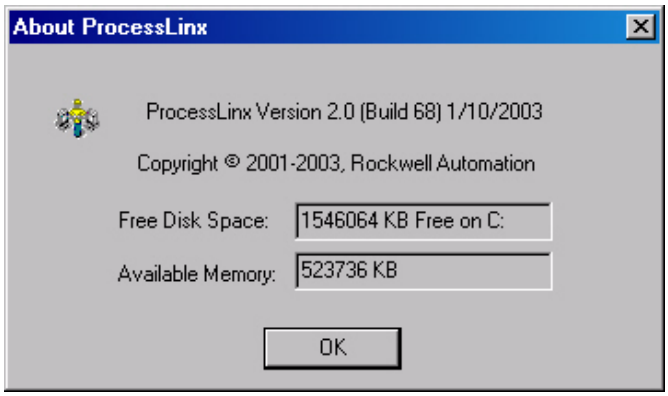


Figure 2.1 ProcessLinx Version Label

Configure Mode

In configure mode, you can modify the ProcessLinx Options and Settings from the main window. ProcessLinx does not communicate with OPC Servers in configure mode.

To change to configure mode from the menu tab, select **Mode ⇒ Configure**. Changing from runtime to configure mode will disestablish all OPC tag items and Groups and stop all communication with the OPC Servers. You can also click

in runtime or press the keys <Shift+F5> to change back to configure mode.

Runtime Mode

To change ProcessLinx to runtime select **Mode ⇒ Runtime**, click

or press <F5>.

While in runtime mode ProcessLinx acts as an OPC Client to both the ProcessLogix server and the external OPC Server. Initiating runtime mode establishes all OPC Groups and Items with the specified OPC Server(s). The opened TagMap configuration (See Figure 2.1) is activated. If an OPC server(s) connection cannot be established upon startup, an error will be logged to the Message Log window and the driver will be placed into configure mode.

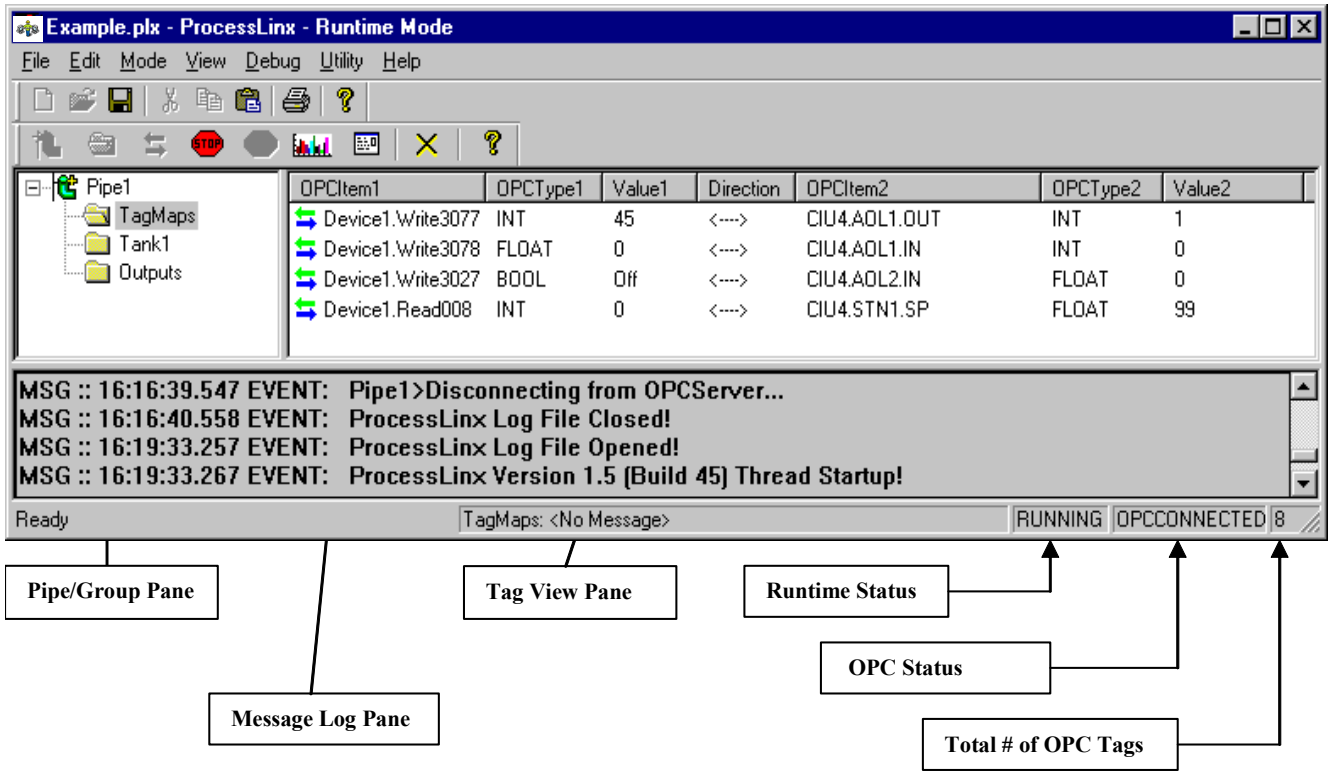

Example.plx - ProcessLinx - Runtime Mode
File Edit Mode View Debug Utility Help
Pipe1
TagMaps
Tank1
Outputs
OPCItem1 OPCType1 Value1 Direction OPCItem2 OPCType2 Value2
Device1.Write3077 INT 45 <----> CIU4.AOL1.OUT INT 1
Device1.Write3078 FLOAT 0 <----> CIU4.AOL1.IN INT 0
Device1.Write3027 BOOL Off <----> CIU4.AOL2.IN FLOAT 0
Device1.Read008 INT 0 <----> CIU4.STN1.SP FLOAT 99
MSG :: 16:16:39.547 EVENT: Pipe1>Disconnecting from OPCServer...
MSG :: 16:16:40.558 EVENT: ProcessLinx Log File Closed!
MSG :: 16:19:33.257 EVENT: ProcessLinx Log File Opened!
MSG :: 16:19:33.267 EVENT: ProcessLinx Version 1.5 (Build 45) Thread Startup!
Ready
TagMaps: <No Message>
RUNNING
OPCCONNECTED
8
Pipe/Group Pane
Tag View Pane
Runtime Status
OPC Status
Message Log Pane
Total # of OPC Tags

Figure 2.2 ProcessLinx Example Window

Configure ProcessLinx Options

To configure ProcessLinx Options select from the Utility menu, **Utility** ⇒ **Configure Options** or click in the tool bar. You see:

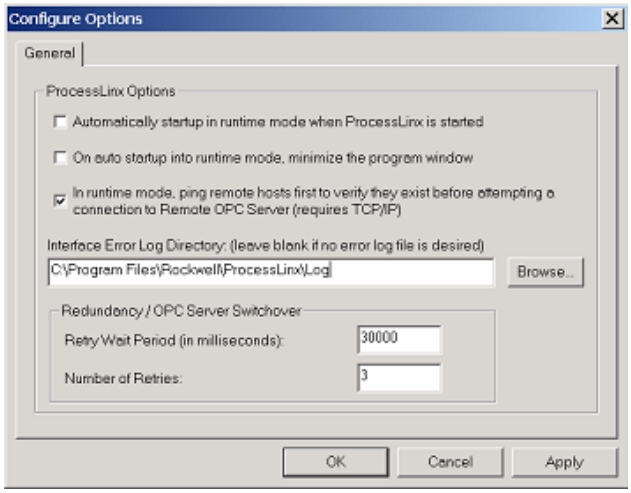

Configure Options
General
ProcessLinx Options
Automatically startup in runtime mode when ProcessLinx is started
On auto startup into runtime mode, minimize the program window
In runtime mode, ping remote hosts first to verify they exist before attempting a connection to Remote OPC Server (requires TCP/IP)
Interface Error Log Directory: (leave blank if no error log file is desired)
C:\Program Files\Rockwell\ProcessLinx\Log
Browse...
Redundancy / OPC Server Switchover
Retry Wait Period (in milliseconds):
30000
Number of Retries:
3
OK
Cancel
Apply

Figure 2.3 Options Dialog Box

Automatically startup in runtime mode when ProcessLinx is started

This option starts ProcessLogix in runtime mode. This change takes effect the next time you start the program. If unchecked then you must manually place ProcessLinx in runtime mode. Default is unchecked.

On auto startup into runtime mode, minimize the program window

This option is used to control whether or not the program starts minimized or not. On startup if the "Automatically startup in runtime mode…" is checked then the program window will be minimized. Default is unchecked.

In runtime mode, ping remote hosts first to verify they exist before attempting a connection to Remote OPC Server (requires TCP/IP)

This option should always be enabled for all remote OPC Server(s) communicating via TCP/IP. If checked, ProcessLinx pings the remote host before attempting an OPC connection to it. If this option is not enabled and

a remote server connection is lost, ProcessLinx may keep trying to connect with it, you may have to restart ProcessLinx.

ATTENTION

This restart situation depends on how you setup DCOM with the COM interface (OPC Server). Be certain to follow the DCOM steps in Chapter 7 for best performance when you configure TCP/IP and DCOM with COM I/F to OPC Server.

For local OPC servers (running on the same PC as ProcessLinx) this option is not used. The default is checked. If enabled, you can change the settings and have them recognized while in runtime operation.

Interface Error Log Directory

Use this option to specify the error log destination directory. If a folder is specified, all error and information messages displayed in the driver's log window are logged. ProcessLinx generates a new log file daily. If this field is left blank, no error or information logging occurs.

Maximum # of Screen Buffer Lines in On-Screen Message Log

Use this option to specify the maximum lines to display in the Message Log Pane. If the current number of lines is greater or equal to this the first 50% of these messages are removed. This only concerns on-screen messages. The messages logged to the log file are not removed.

This option also minimizes memory usage. The buffer can consume up to 1 MB of memory otherwise.

Clear On-Screen Messages When Limit is Hit

Use this option to specify when the number of lines displayed in the Message Log Pane hits the limit defined in the option above. The whole buffer (all on-screen messages) is cleared. If unchecked then only the first 50% will be removed.

Retry Wait Period (in milliseconds) for Redundancy / OPC Server Switchover

Use this option to specify the retry wait period before attempting to connect to an OPC Server that has stopped communicating. This is used in runtime mode during a switchover. Changes to this setting can be made, and are recognized during runtime operation.

Number of Retries for Redundancy / OPC Server Switchover

This option specifies the number of retries attempts to connect to a redundant OPC Server. Retries are initiated after communication is lost with a server. For example, if this is set to three and communication is lost with the primary server, ProcessLinx will switchover to the secondary server (if defined). Three connection attempts are made to the secondary server before switching back to the primary. ProcessLinx then attempts a connection three times with the primary and so on. Changes to this setting can be made, and are recognized during runtime operation.

Notes:

ProcessLinx TagMap Configuration

A ProcessLinx TagMap configuration is created to communicate OPC items (also known as tags) from one OPC server to another. This configuration maps each data point exchanged between systems using the OPC protocol.

The ProcessLinx configuration is a hierarchy of user-defined objects. The top-level object is a Pipe. A Pipe defines which OPC Servers are used for data exchange. It also defines the primary and secondary PC (if available) hosts of the external OPC Server(s). The next level down is called a Group.

Each Pipe contains Groups. Groups contain TagMap definitions. Each Group object can contain as many TagMaps as needed, collectively "Grouping" similar tags together. The Group is conceptually similar to a file folder. All TagMaps in a Group have the same data update rates, dead band, time bias and data flow direction for all TagMaps. During runtime, each defined ProcessLinx Group creates and encapsulates an OPC-standard Group.

A TagMap defines the source and destination OPC Items for each data point. Each map specifies the actual OPC Item names, OPC Server Access paths, data flow/data master, and scaling options. In runtime, the TagMap object contains an OPC-standard Item object for each server.

Creating a ProcessLinx Configuration

IMPORTANT ProcessLinx will not load or open a configuration in runtime, the interface must be placed into configure mode first.

To:	Select:
create a ProcessLinx configuration	**File** ⇒ **New…** An empty Pipe called Pipe1 displays in the tree view of the main window.
save, copy or backup the configuration	**File** ⇒ **Save** (or) **Save As…** and give it a meaningful name. The default ProcessLinx TagMap configuration filename is *.plx file extension.

Opening a ProcessLinx Configuration

To:	Select:
open an existing ProcessLinx configuration	**File** ⇒ **Open...** The file is loaded into the tree and tag view panes. The configuration can be edited or placed into runtime mode. When you have large taglists it takes ProcessLinx longer to start up and load the lists into memory.
edit an object such as a Pipe, Group or TagMap	**File** ⇒ **Properties...** In runtime mode you can edit a limited set of properties as well as save a ProcessLinx configuration.

Creating a Pipe

To:	Select:
create a Pipe	**Edit** ⇒ **Add New Pipe...** or by pressing <Ctrl+P>
configure the Pipe	**Edit** ⇒ **Properties…**

Using the Pipe Properties dialog box in Figure 3.1, define the parameters for the ProcessLogix Server and the External OPC Server.

Redundant ProcessLogix or External OPC Servers are supported if you provide the remote node information in the respective Secondary Host Name edit box. If redundancy is not required, leave the Secondary Host

Name blank or set it to “localHost”.

Included in ProcessLinx 2.0 is an executable that only allows you to create and/or Import up to 10 Pipes. Configure this setting by creating/editing a REG_DWORD Registry key value at:

HKLM\SOFTWARE\Rockwell Software\ProcessLinx\Options\MaxPipes

The default is 10 if the Key is not found. If value is set to less than 1 the default is set to 10 since there should always exist at least 1 Pipe. Importing more than 10 Pipes and creating more than 10 Pipes has been tested and in each case a Message Box will prompt you to the actual limit allowed.

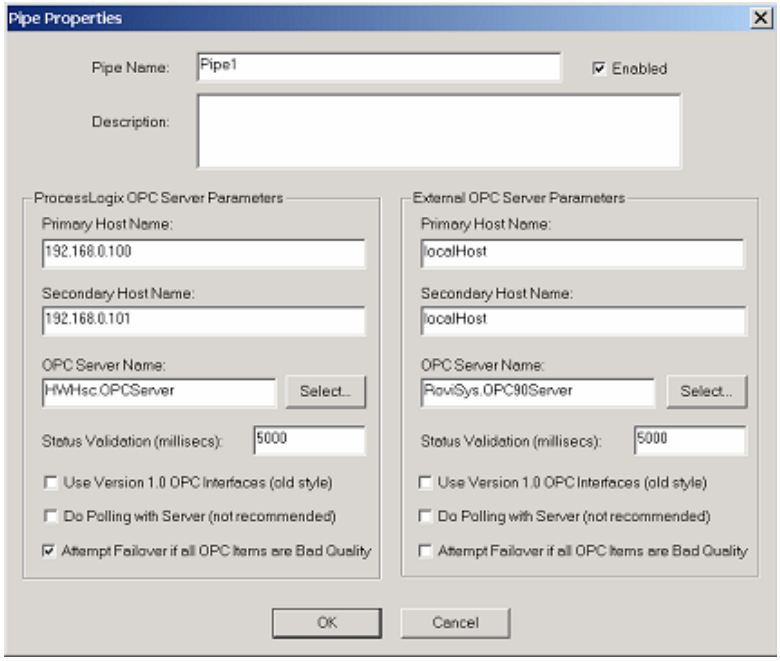


Figure 3.1 Pipe Properties Dialog Box

Using the Pipe Properties dialog box

1. Select a meaningful name for the Pipe.

2. If needed, enter comments about the interface in the text description box.

3. Enter the host name(s) and OPC Server Name(s) for the ProcessLogix and the External OPC Server for the new Pipe.
   a. To browse for an OPC Server click **Select…**
   A "Select OPC Server" dialog box containing all installed OPC Servers on the local machine will be displayed.
   b. Select the desired external OPC Server and click OK.

IMPORTANT

If the OPC Server does not show up in the list, then it has not been properly installed or registered on the machine.

Remote servers do not have to be installed on the local node in order to connect with them. Their OPC Class information (GUID, refer to Preparing the Client's User Accounts on page 1-3) needs to be registered on the local machine. Consult the specific OPC Vendor for documentation on how to register an external OPC Server to run on a remote node.

If documentation is not provided, obtain a copy of the executable of the external OPC Server and run it once on the local machine. Most OPC Servers will register themselves automatically when they are executed for the first time.

Enable

The enable checkbox enables or disables the Pipe and all of its children (Groups and TagMaps) in runtime from communicating with each other. Be certain to check this if you want the Pipe to run in runtime mode, otherwise it will be disabled.

This is useful when multiple pipes are configured, and one for each of several external servers, and at some point, one of the OPC servers or remote system is brought down (for maintenance, for example) and you don't want or need that pipe interfacing with the downed OPC Server. In this case you would simply uncheck the enable checkbox for that pipe. Otherwise, if left enabled, it may take away processing power from the interfaces on the other pipes due to time-outs and DCOM problems.

Status Validation

The Status Validation edit box specifies the interval for checking or validating the OPC Server. ProcessLinx validates the status of the primary and secondary external OPC servers at the rate specified by this dialog. ProcessLinx uses this status to determine if a fail over to the redundant ProcessLogix Server is required. It is specified in milliseconds with a default of 5000.

Use v1.0 OPC Interfaces

The Use v1.0 OPC Interfaces checkbox indicates that v1.0 OPC Interfaces are used when reading/writing data with the OPC Server. This option is for OPC Servers that do not support the v2.0 Interfaces. By default this option is left unchecked. ProcessLinx will attempt to use the v2.0 Interfaces if left unchecked, and will automatically revert to the v1.0 interfaces if the Server does not support v2.0. This option allows for the explicit selection of the 'old-style' v1.0 interfaces. If this box becomes checked during runtime, then ProcessLinx found that the OPC Server did not support v2.0 which may be due to a DCOM security issue. Refer to Quick Setup on page 6-1 for more information.

Do Polling with Server (not recommended)

The Do Polling With Server checkbox enables polling of the OPC Server at regular intervals. The Group's Group Rate specifies the polling interval. Polling the Server implies that ProcessLinx will use the IOPCSyncIO interface to read all items from cache at the poll interval. This option is not recommended. If v2.0 Interfaces are working correctly then there should be no need to read the data synchronously. This option is provided for OPC Servers that are setup incorrectly and/or do not support advise subscription mechanisms.

OPC Advise is provided under v1.0 with the IadviseSink Interface and under v2.0 as the IOPCDataCallback using a ConnectionPoint. If this box becomes checked during runtime, then ProcessLinx found that the OPC Server did not support v1.0 Advise which may be due to a DCOM security issue or other problems with COM. Refer to Quick Setup on page 6-1 for more information.

Do Polling with the Server: Recommended Number of Tags

Placing an absolute limit on the number of tags is difficult since there many dependencies, such as Processor speed, memory, OPC Server type, OPC Server throughput, connection type (serial or Ethernet). 300 tagmaps is approximately the limit.

ProcessLogix throughput is about 280-320 values per second, so any more tags per second that ProcessLinx has to poll, it becomes theoretically impossible. Polling is provided as an option as a 'last resort' effort if callbacks do not work or there is a problem (defect) in the OPC Server. It is not recommended that the user select this option.

Attempt Failover if all OPC Items are Bad Quality

This checkbox will enable or disable failover when all OPC Items are bad Quality. This option is specifically used with ProcessLogix redundant Server scheme but can be used with any OPC Server. The redundant (backup) OPC Server of ProcessLogix will set all OPC Items to 'Out-of-Service' BAD Quality when the node becomes the backup and thus this can be used as a trigger for ProcessLinx to failover to the redundant server. Failover for other reasons (connection time-outs, server quit) will still automatically occur if the secondary server is defined.

Successful Failover from one OPC90 Server to Another

A successful failover occurs when the active connected server [Primary/Secondary] 'fails' and ProcessLinx then connects to the configured redundant one and after which the redundant one remains 'Connected' and thus is not 'failed'. 'Server Failure' is defined below.

What Triggers a Failover

'Connected to' means that ProcessLinx can establish a link to the configured server's OPCServer interface, add groups, add items and have data callbacks setup. In runtime, status of the server is checked at the periodic rate specified in the Status Validation edit box for the server in the Pipe dialog box. The GetStatus( ) function on the OPCServer interface is used to provide an initial decision as to whether processLinx is connected to the server. The following are conditions that move ProcessLinx to implement a failover (other than the OPC Server failing).

- If the GetStatus( ) or Write( ) functions fails 5 times (meaning the functions return a server level error or Win32 error of RPC_C_SERVER_UNAVAILABLE) or the server status is returned as (FAILED, NOCONFIG or SUSPENDED) then the server is marked as 'failed' and if a redundant server is configured then ProcessLinx will attempt to connect to that.

- If the 'Attempt failover of all OPC Items are Bad Quality' is checked in the Pipe dialog box, then all OPC Items of every group for that server are checked as well at the periodic rate specified in the Status Validation edit box in the Pipe dialog box.

- If all active tags in any active group in that server are BAD Quality and the sub-status quality is either CONFIGURATION_ERROR or OUT_OF_SERVICE then that server is marked as 'Failed'.

- If just one of those active tags is GOOD or QUESTIONABLE quality then server is not failed on the basis of 'Bad Quality' since data can be obtained for at least one of the items. This also means that the server cannot be 'failed' on the basis of Bad Quality if the user set all groups in that server to Inactive (i.e. No active groups) since there will be no active tags to check quality with.

Creating a Group

To:	Select:
create a Group	1. Select the Pipe 2. Select **Edit** ⇒ **Add New Group...** from the main menu or use <CtrlG>. The "Add Group" function is also available when you right mouse click on the Pipe, via a pop-up menu.

The Group Properties dialog box in Figure 3.2 displays.

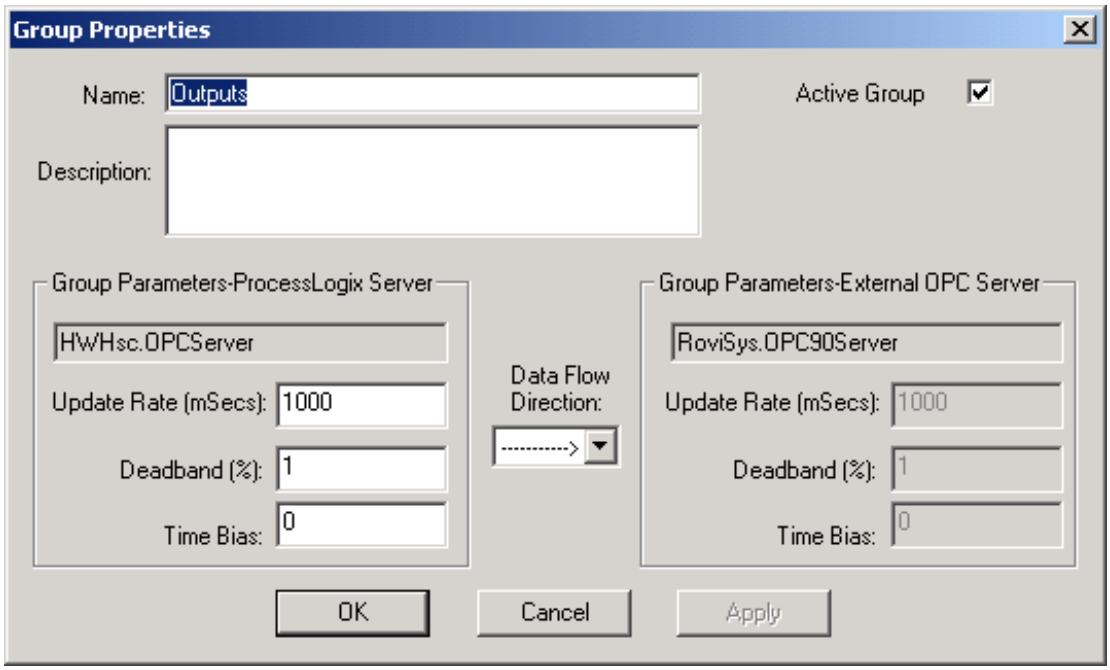


Figure 3.2 Group Properties

The Group Properties dialog box allows specification of the OPC Group parameters. These parameters affect all TagMaps and OPC Items contained in the Group. Define a descriptive name for the new Group. You can also enter a Group description.

The number of Groups you can add is limited by memory and possible OPC Server limitations. The OPC Server may set a limit on the number of groups you can define, if the limit is exceeded then the error CONNECT_E_ADVISELIMIT is returned and ProcessLinx will display a message ("the server hit the Advise limit, please limit the # of groups defined") that is logged in the log file.

Data Flow Direction

IMPORTANT Data Flow Direction is the only parameter that cannot be changed in runtime.

The available data flow directions between the ProcessLogix OPC Server and the External OPC Server for all TagMaps within the Group are:

→ Specifies that OPC items are read from the ProcessLogix Server and written to their map in the External Server.

↔ Specifies that data flows in both directions. The data is read or written to the ProcessLogix OPC Server and written or read from External OPC Server.

← Specifies that OPC items are read from the External OPC Server and written to their maps in the ProcessLogix OPC Server.

TIP

The Group parameters, described below, can be changed in runtime mode. The new settings can be applied immediately by selecting **Apply** or **OK**.

Table 3.A Group Parameter Description

Parameter	Description
Active Group	This checkbox specifies the Group parameter's active status. Data for all TagMaps in the Group can only be transmitted if the Group is active within the OPC Server in runtime. Group communication is disabled by unchecking this option. This can be modified in runtime to enable or disable particular Groups. A disabled (Inactive) Group will neither receive or transmit values for any items defined within it. All tags within an Inactive Group are still established in runtime with the Server. You can set a Group inactive in runtime to disable updates for all the Group's items.
Update Rate (mSecs)	This Group parameter specifies the update rate (in milliseconds) for the OPC Group and is specified to the OPC Server when the OPC Group is created in runtime mode. The rate tells the OPC server how often to receive data or check for changes in this Group's OPC Items. Most OPC Servers are exception-report based, meaning they will send data only when it changes. However, some OPC Servers will send data at this rate even if it does not change. This option can be changed in runtime mode.
Deadband (%)	This Group parameter specifies the deadband (in percentage of value) of an OPC Item. Most OPC Servers use this parameter to determine when to notify a client of a change in value. If the value is noisy, you may not want the OPC Server sending each value unless it changes by more than this percentage.
Time Bias	Time bias offsets the time stamp of each OPC item by the specified amount. This parameter is required by the OPC standard for creating OPC Groups within an OPC Server. It is designed to compare time stamps of the OPC items that may exist in different time zones. However, most OPC Servers do not utilize this parameter. ProcessLinx neither uses nor transfers the time stamp date. Time Bias is available for future implementation. The Default is 0.

Creating a TagMap

TagMap definitions are created within a parent ProcessLinx Group. These definitions relate OPC Items from an external OPC Server to OPC Items in the ProcessLogix OPC Server. A tag within the ProcessLogix Server consists of the Point ID and parameter name (e.g. TIC101.PIDA.SP). A tag within the External OPC Server consists of its specific tag or Item ID representation (i.e. DEVICE, BLOCK and PARAMETER name).

To create a TagMap:

1. Select the desired Group.

2. Select **Edit ⇒ Add TagMap...** from the main menu or right mouse clicking on the Group and then selecting "Add Single TagMap". The TagMap Definition dialog box, (shown below) displays for the TagMap and can be edited.

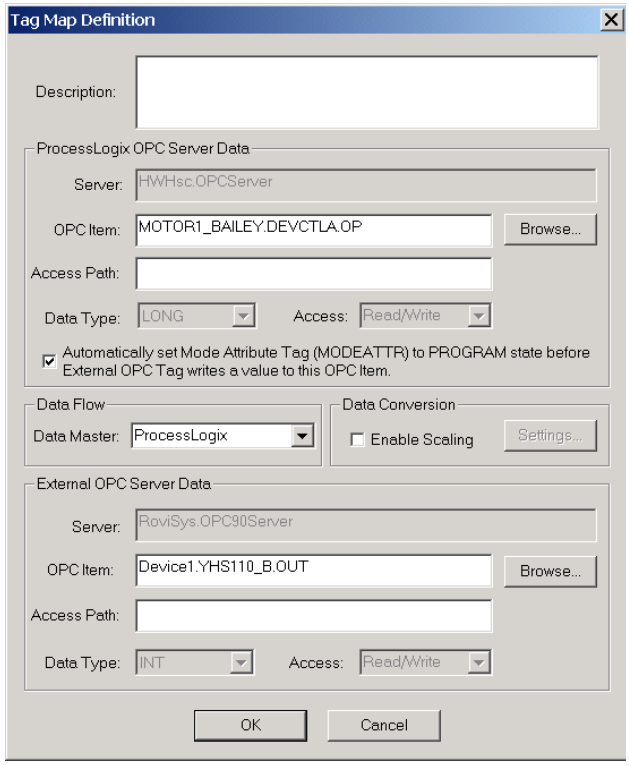


Figure 3.3 Definition Dialog Box

Attributes are specified as part of defining the TagMap in the above dialog box. The TagMap parameters are defined below:

Table 3.B TagMap Parameters

Parameter	Description
Description	You can enter a description for the TagMap entry. This field can be left blank if desired. **IMPORTANT:** Commas are not allowed in the description.
Data Flow / Data Master	Data Master defines where the data originates when starting runtime mode. This selection is only enabled when the Group's "Data Flow Direction" attribute is set to both directions (←→). Valid entries are "ProcessLogix" or "External OPC Server". Starting runtime mode for bi-directional Groups produces tag data from both OPC Servers. The Data Master entry eliminates any ambiguity by initializing the Group's data using the selected OPC Server as the source.
Data Conversion	Data Conversion/Scaling provides a method for translating data. The conversions are stored in Translation Tables. Refer to to the section Data Conversion / Scaling for details about creating and modifying translation tables.

IMPORTANT Each of the attributes described in Table 3.C must be defined for both OPC Servers in the TagMap.

Table 3.C TagMap Data Attributes

Attribute	Description
Server	The Server field displays the defined OPC Server for this half of the TagMap.
OPC Item	Use the OPC Item area to enter the specific tag in the OPC Server. For OPC v2.0 compliant servers, "Browse" opens the database for point-and-click OPC Item entry. When the Item is selected, its tag name is entered in thus field. **Tip:** To browse OPC Items from ProcessLogix, export the Control Builder configuration. Refer to Creating a ProcessLinx TagMap Configuration on page 3-15. If you want to only monitor an item from a Server then it is not necessary to define the OPC Item for the other Server, leave the OPC Item field blank. In this case, the value of the undefined item displays as <Invalid Tag> in Monitor mode. OPC items can be added/edited in runtime.
Access Path	Access Path sets the OPC path for the specific OPC Item. For ProcessLogix OPC Server and most External OPC Servers this can be left blank. If this parameter is implemented within the External OPC Server, consult it's documentation for specifics. For example, the access path may be COM1 or COM2.

Table 3.C TagMap Data Attributes

Attribute	Description
Data Type	Data Type displays the default OPC tag type for the selected OPC Item. This is set to "native" initially. In runtime, the actual OPC Item's data type is received from the OPC Server and subsequently displayed.
Access	The OPC Item Access displays in this field. This attribute is defined in the Parent Group and is read only. **"Read Only"** indicates that no value can be written to the OPC Item. **"Write Only"** means the OPC Item is written to and not read. Exception reported values from the OPC Server are disregarded if the Item is Write Only. **"Read/Write"** indicates the tag is both read from and written to. In this case, the Data Master entry defines which OPC Server initializes.
Automatically set Mode Attribute Tag (MODEATTR) to PROGRAM state	This ProcessLogix-Only selection allows proper mode conversion between an External OPC Server and the ProcessLogix Server. If the ProcessLogix OPC Item is a control element (such as an OP, SP or MODE) that is written to from the External OPC Server, this must be enabled. In order for an external source to change the value of an OP, SP or MODE element in ProcessLogix, it needs to set the MODEATTR tag to PROGRAM state first and then write the value. Otherwise ProcessLogix will reject the value. By setting this checkbox ProcessLinx automatically sets the mode to "Program" and writes in the new value. If the Server is not a ProcessLogix system, the checkbox does not appear.

Creating a ProcessLinx TagMap Configuration

Once you create the ProcessLogix and the External OPC Server configuration a ProcessLinx configuration is created.

ProcessLogix Control Points defined within Control Builder can be browsed. Other points such as Analog, Accumulator and Status Points cannot. To make the ProcessLogix points browseable, select the desired control blocks from Control Builder and select **File ⇒ Export**. This operation exports the selected control blocks to UNICODE ASCII files which ProcessLinx parses and compiles into a browseable list of function blocks for the Tag Maps dialog box. ProcessLinx uses this compiled list of blocks in conjunction with an OPC item file. The item file <ProcessLinx.itm> defines all the parameters or OPC Items for each block type found in the compiled list of function blocks.

(This item file should be provided by Rockwell Automation and exists within the same directory as the ProcessLinx executable.)

By default, ProcessLinx attempts to use the item file when browsing ProcessLogix tags used to map tags. If this file does not exist then each time the "Add TagMap(s)" function is executed it parses and compiles all exported function block files and creates the function block list with no connectable items to map to. If this happens, create a ProcessLinx.itm file manually by performing the following steps:

1. Run Control Builder and export the desired blocks using the File ⇒ Export function from the main menu in ControlBuilder.

2. Stop and exit ProcessLinx.

WARNING

Rockwell Automation holds no responsibility for damages to property and personnel or loss of time that may occur when making changes to the Registry.

3. Run the Registry Editor program <Reedited> from the **Start ⇒ Run** dialog box.

4. Create and set the following keys to specified values under the branch:

<HKEY_LOCAL_MACHINE\SOFTWARE\RockwellSoftware\PrcessLinx\Options>

Table 3.D Registry Key

Registry Key to create	Key Type	Set the value of this key to:
ExportBrowseFileName	STRING	C:\Program Files\ProcessLogix\ProcessLinx.itm or set the path to the location of ProcessLinx.exe. This setting should match the <DefaultBrowseFile> string value.
ExportBrowseFile	DWORD	1

5. Start ProcessLinx and select **Add Tag Map(s)**. ProcessLinx creates the specified item file to browse available points if you have defined the HWHsc.OPCServer for the server.

6. Stop and exit ProcessLinx.

7. Delete only the 2 created registry keys above. Do not delete the <DefaultBrowseFile> registry entry. If these 2 keys are not deleted, the item file will be created each time the **Add Tag Map(s)** function is executed.

8. Start ProcessLinx again.

9. Execute the **Add Tag Map(s)** function and the available items appear

under the function blocks. Note: Some of the items are not connectable.

Data Conversion/Scaling

Most analog values exchanged between the ProcessLogix OPC Server and an External OPC Server are already expressed in engineering units and therefore, linear scaling of analog values is not required. To accommodate discrete values not meeting this condition, TagMap definitions allow scaling.

OPC Items

ProcessLogix OPC Items are accessed from one of four point types/categories: Status points, Analog points, Accumulator points and Control Processor points. The ProcessLogix OPC data item addressing is:

Simple Points

Simple Points are those other than Control Processor points.

- point.parameter [/enum]

Control Processor Points

- point.block.parameter [/enum]

Table 3.E Data Addressing Terminology

Term	Description
point	point ID
parameter	point parameter name listed in "Point Parameter Data Addressing" section in Knowledge Builder. Navigate to Knowledge Builder ⇒*Server and Client Configuration Guide* ⇒*Using ProcessLogix Data in Other Applications* ⇒*ProcessLogix OPC Support*.
block	control module parameter for Control Processor Points.
/enum	– specifies whether the parameter should be treated as an enumerated type. Defining this after the Item name returns the string enumerated value for the point (i.e. MAN or AUTO for a controller mode point).

Some OPC Servers do not support string value writes from OPC Clients, such as ProcessLinx. The "/enum" designation should be used carefully.

Some of the main parameter names along with their data types for each ProcessLogix point type are given below.

TIP

Refer to the *ProcessLogix Server and Client Configuration Guide* documentation in Knowledge Builder for a complete listing of all supported data types. Refer to the *ProcessLogix Parameter Reference Manual* in Knowledge Builder for parameter definitions.

Table 3.F Status Point Parameters

OPC Server parameter name	OPC Type	OPC Access
PV	VT_I4	R/W
Mode	VT_I4	R/W
OP	VT_I4	R/W
PVInError	VT_BOOL	Read only
ModeInError	VT_BOOL	Read only
OutputInError	VT_BOOL	Read only

Table 3.G Analog Point Parameters

OPC Server parameter name	OPC Type	OPC Access
PV	VT_I2	R/W
Mode	VT_I4	R/W
OP	VT_I2	R/W
SP	VT_I2	R/W
EULO	VT_R4	Read only
EUHI	VT_R4	Read only
PVInError	VT_BOOL	Read only

Table 3.G Analog Point Parameters

ModeInError	VT_BOOL	Read only
SetPointInError	VT_BOOL	Read only
OutputInError	VT_BOOL	Read only
InLowAlarm	VT_BOOL	Read only
InHighAlarm	VT_BOOL	Read only
PointInAlarm	VT_BOOL	Read only

Table 3.H Accumulator Point Parameters

OPC Server parameter name	OPC Type	**OPC Access**
PV	VT_I2	R/W
EUHI	VT_R4	Read only
RawPVValue	VT_I4	R/W
PVInError	VT_BOOL	Read only

ProcessLogix Control Point parameters are dependant upon the type of control point or function block being referenced. The following tables display only the control points used most frequently in interfacing to other OPC Server blocks.

Table 3.I Table 7.D Control Point Parameters for a PID or PIDA block

OPC Server parameter name	OPC Type	**OPC Access**
PV	VT_I4	R/W
MODE	VT_I4	R/W
OP	VT_I4	R/W
SP	VT_I4	R/W
PVEULO	VT_R4	Read only

Table 3.I Table 7.D Control Point Parameters for a PID or PIDA block

PVEUHI	VT_R4	Read only
K	VT_R4	
KLIN	VT_R4	
K	VT_R4	
T1	VT_R4	
T2	VT_R4	
CVEULO	VT_R4	Read only
CVEUHI	VT_R4	Read only
OPHILM	VT_R4	
OPLOLM	VT_R4	
SPHILM	VT_R4	
SPLOLM	VT_R4	

Table 3.J Control Point Parameters for a Device Control DEVCTL Function Block

OPC Server parameter name	OPC Type	**OPC Access**
PV	VT_I2	R/W
MODE	VT_I2	R/W
OP	VT_I2	R/W
MODEATTR	VT_R4	Read only

Examples: HS_036.DEVCTLA.PV or pidloop.PIDA.SP.

Translation Table

To enable scaling for a particular TagMap, you must define and select a translation table. In runtime, the table converts values from one OPC Server to valid values for the other OPC Server.

This is useful for converting the discrete mode states of a ProcessLogix DEVCTL function block an External OPC block. You can define as many translation tables as needed. The correct translation for the TagMap is selected from the available list. To define a translation table, open the TagMap Dialog box (for the TagMap) and click "Settings…".

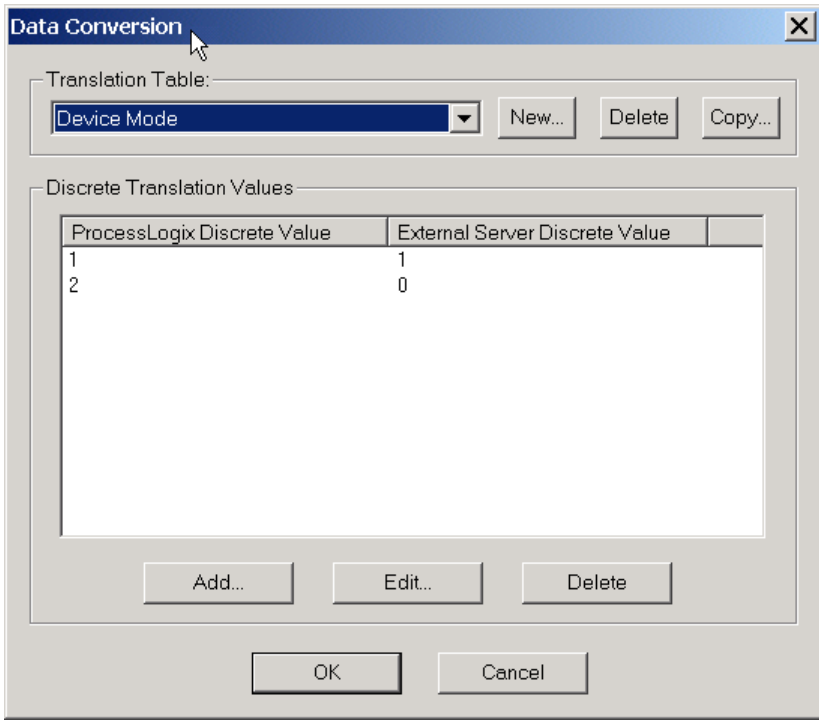


Figure 3.4 Data Conversion Dialog Box

- To add a new conversion table, click **New**… and enter a descriptive name for the table.
- To delete a Translation table select the table and click **Delete**.

IMPORTANT If any TagMaps are referencing the table, it cannot be deleted.

- To copy a translation table, select the translation table and click **Copy**. Enter a new descriptive name for the new table.

- To add or edit a new translation value map for the selected translation table, select **Add**.

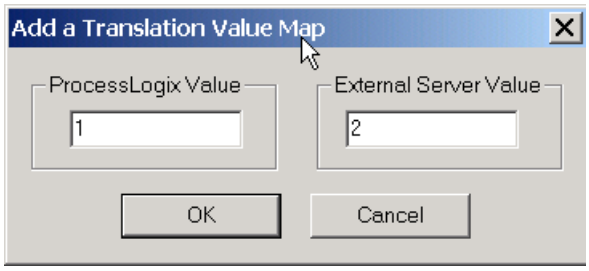


Figure 3.5 Translation Value Dialog Box

Enter the given conversion values for either side of this translation. Repeat the Add... step for all discrete values.

TIP

Translations can be added and edited using CSV files.

Refer to Import/Export on page 5-1 for more information.

The example in Figure 3.4 shows two ProcessLogix discrete values mapped to an External OPC Server. In ProcessLogix, the enumerated values for auto and manual are 1 and 2 respectively, 1 and 0 in the external system. This map sets the ProcessLogix Auto Mode Value (1) to the External Value (1). It also sets the ProcessLogix Manual Value (2) to the External Manual Value (0).

After the Translation table and value maps are setup, click OK and the TagMap will be set to use this translation table. If o conversion map is defined, the value is sent as-is to the other OPC Item defined in the TagMap.

Editing a Pipe, Group or TagMap

To edit a Pipe, Group or TagMap simply double-click on the object or select the object and then select **Edit ⇒ Properties…** from the main menu or right mouse click on the object and select "Properties…" from the pop-up menu. Most object properties can be edited in runtime. ProcessLinx supports changing an OPC Item reference in runtime.

Copying/Moving TagMaps between Groups

In configure mode, one or more TagMaps can be copied or moved from one Group to another. Select the TagMaps to copy or move and drag them to the desired Group. You can select Multiple TagMaps by holding down the Control or shift keys.

If you hold down the <Control> key before you drop the object(s), the objects are copied to the desired Group. Otherwise the TagMaps are moved to the new Group.

Multiple TagMap Configuration

Multiple tags can be mapped quickly using the "Map Tags" dialog box. Highlight the desired Group and Select **Edit ⇒ Add TagMap(s)...** from the main menu or press <Ctrl T> or right mouse click on the Group and select **Add TagMap(s)…**.

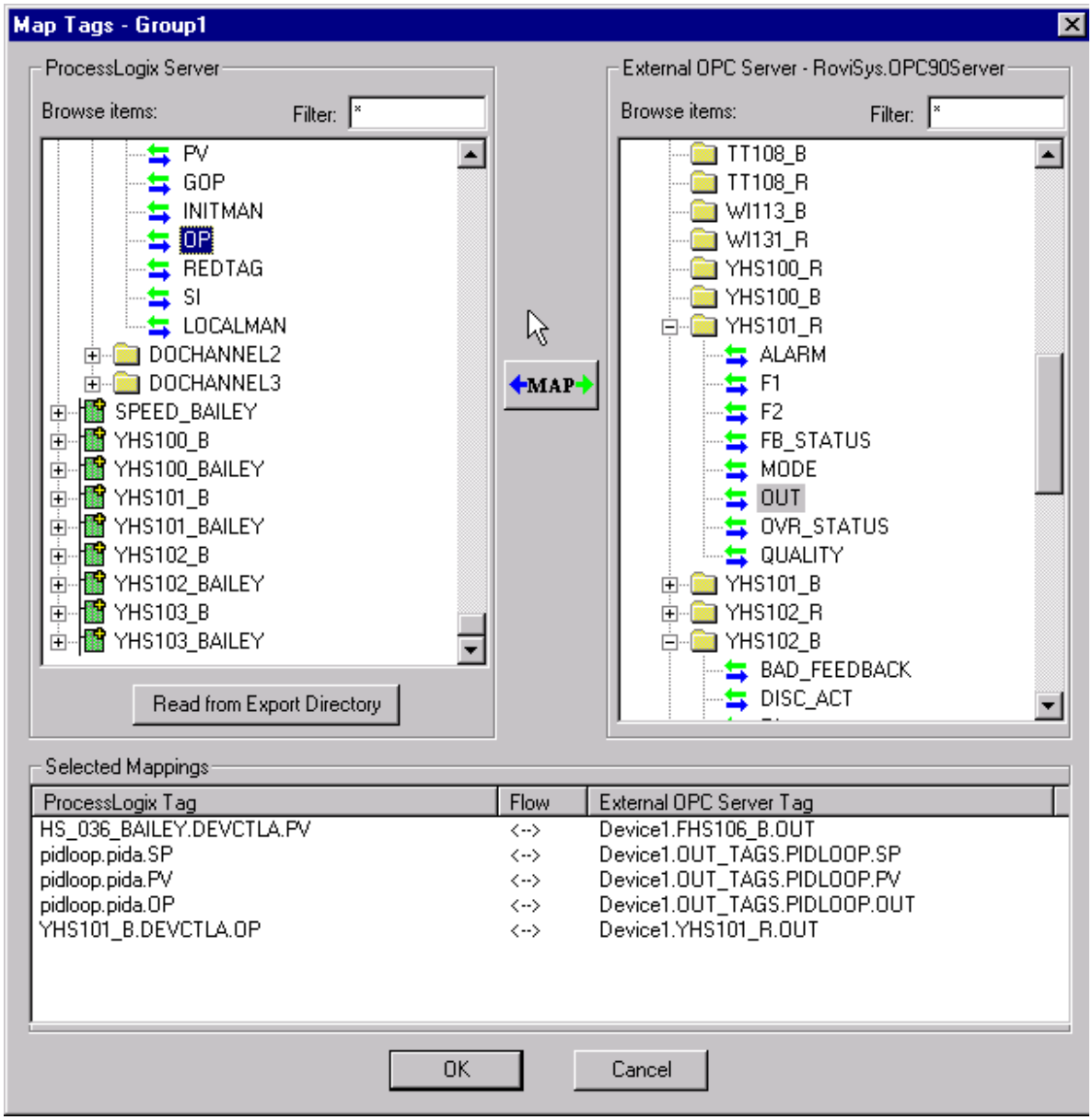


Figure 3.6 Multiple TagMap Dialog Box

This dialog box displays two tree views. To enable this view and have

browse capability the External Server must be registered on the local PC. Each pane contains a tree view listing of all available OPC items within the Server.

You select an OPC item on one side and the corresponding item to map to on the other and click ←MAP→. Alternatively, the user can select an item from one side and drag it into an OPC Item on the other to create a tag mapping.

The data flow direction is decided by the direction of the Parent Group. The tags mapped during the current session are displayed in the "Selected Mappings" window. Selecting cancel, clears all selected maps for the current session.

To remove a TagMap from an edit session, select the TagMap you want to delete in the "Selected Mappings" pane and press Delete. Clicking OK accepts the new maps (if any) and loads them under the current Group. TagMaps can be added in runtime mode.

In ProcessLogix R500 systems, it may be necessary to define groups of tags within ProcessLogix into Areas. The Areas will display in the OPC Server address space.

In ProcessLogix R400 systems, the Top Level (root) Node in the Browse Tree was displayed with an asterisks and all items could then be defined from this top level name (such as *.FC151.PIDA.SP). ProcessLogix R500 systems still support the use of the * as a prefix on an OPC Item but are now displayed under the defined area.

Testing a ProcessLinx Configuration

A ProcessLinx configuration can be tested after the above steps are implemented. It is recommended at the initial onset of testing the ProcessLinx interface that you set the debug mode to 5, turn on the Show flags (i.e. Show Errors, Show Events and Show Send), and specify an error log directory to view any errors. You can use the **View ⇒ Monitor** function to view the data tag values in real time while testing.

After all TagMaps have been successfully setup and are working according to the Interface Control Strategy in step 3.3, turn off the debug flags with the exception of the Show Errors flag.

ProcessLinx Runtime Operation

Each Pipe created in ProcessLinx provides a conduit to pass information between OPC Servers. An OPC Server cannot communicate directly to another OPC Server. Each OPC Server expects responses that are normally received from an OPC Client. ProcessLinx uses OPC Client responses to receive data from one OPC Server and transmit it to another OPC Server.

When runtime mode is started, the Groups defined in each ProcessLinx Pipe are created in the respective OPC Servers. Then, all OPC Items defined by the Groups' TagMaps are added. Data is transferred from OPC Server to OPC Server based on the Group's settings. The OPC Groups and Items in disabled Pipes are not connected to in runtime. The Enable checkbox is found in the Pipe Properties Dialog.

Monitoring Values

In runtime, monitor OPC Items by selecting **View ⇒Monitor** from the main menu. Current values appear under the *Value1* and *Value2* columns in the tag view pane. If the Server or Group is not connected (or is disabled), the values are displayed as "Uncertain" or "Bad". Monitored values are refreshed every 100 milliseconds. Select another Group to view its TagMap values.

IMPORTANT Groups that have "write only" data flow may not present the latest available data from the OPC Server. The displayed value is the last value written to the OPC Server. No OPC Server "reads" are executed for these Group types.

TIP

If the tagmap displays with a **!**, then one or both of the OPC Items defined within that tagmap are invalid. This means that the OPC Item does not exist within the Server or is mis-configured in the ProcessLinx with an incorrect item name.

Reading OPC Values

ProcessLinx allows any OPC Item to be read during runtime. Select and right mouse click on the desired TagMap's OPCItem1 column in Select **Read ⇒ OPCItem1 or OPCItem2** from the corresponding pop-up menu.

Writing OPC Values

In runtime, values can be manually written to an OPC Item. Select and right mouse click on the desired TagMap's OPCItem1 column. Select **Write ⇒ OPCItem1 or OPCItem2** from the corresponding pop-up menu.

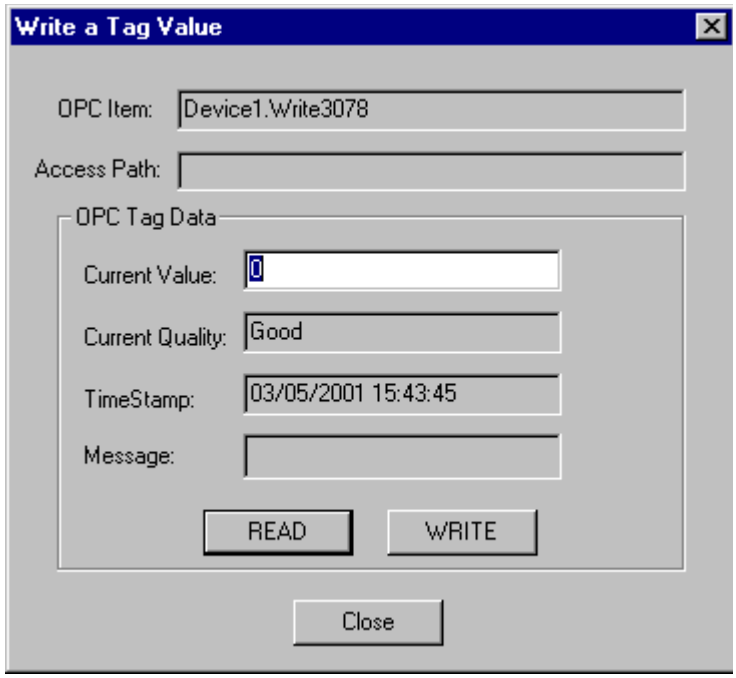


Figure 4.1 Writing to an OPC item

Enter the new value in the Current Value field. Click **WRITE** to force an OPC write to the specified OPC Item. To verify the value was correctly written, click **READ**. WRITE is disabled if the Group data flow property for this Item is set to "read".

TIP

A '?' displayed for Current Value and/or Current Quality indicates that the value for the OPC Item is suspect. This could be the result of an invalid tag reference or a disconnected OPC Server.

The Write Tag dialog box helps debug OPC Item problems. If the OPC Server for this item has an available sub-status message, it displays in the Message field.

Redundant OPC Servers / Fail over

ProcessLinx supports redundant OPC Servers using networked hosts. The Primary and Secondary hosts are configured in the Pipe dialog box. When runtime mode is started, each Pipe connection is attempted with the defined primary host. If that connection attempt fails, a new attempt is sent to the secondary host. If both attempts fail, ProcessLinx control mode is set to Configure for the user to correct the situation.

After startup, if communication to an OPC Server fails, ProcessLinx implements a "cold fail-over". In this case, ProcessLinx disconnects all OPC links to the "failed" Server and attempts a switch to the redundant Server.

In the first step of the fail-over, ProcessLinx verifies that a redundant Server is configured. A network ping is sent to the new remote node to verify its availability. A connection attempt is made to the specified OPC Server on that node. If the connection attempt succeeds, the switchover is complete. If it does not succeed, ProcessLinx attempts to re-connect for the "Number of Retries" specified in the **Utility ⇒ Configure Options** menu.

After the number of retries has been exceeded, ProcessLinx attempts to fail over to the first Server. This connection attempt / fail over process continues until a successful connection has been established. Messages for each failed attempt are logged to the ProcessLinx message pane and error log. Refer to What Triggers a Failover on page 3-7

For redundancy to work properly, OPC Items must be configured identically in the redundant OPC Server pair. If an OPC Item called "Device1.PV1" exists in the primary OPC Server's configuration, an OPC Item called "Device1.PV1" must also exist in the secondary OPC Server's configuration. Missing OPC Items will fail and display Invalid OPC Item message. The OPC Server still indicates running, but one or more OPC Items may not have valid data.

Server Status

You can view the runtime status of each Pipe's OPC Servers. Select **Utility ⇒ Server Status….** Or right mouse click on a Pipe or Group and click **Server Status**.

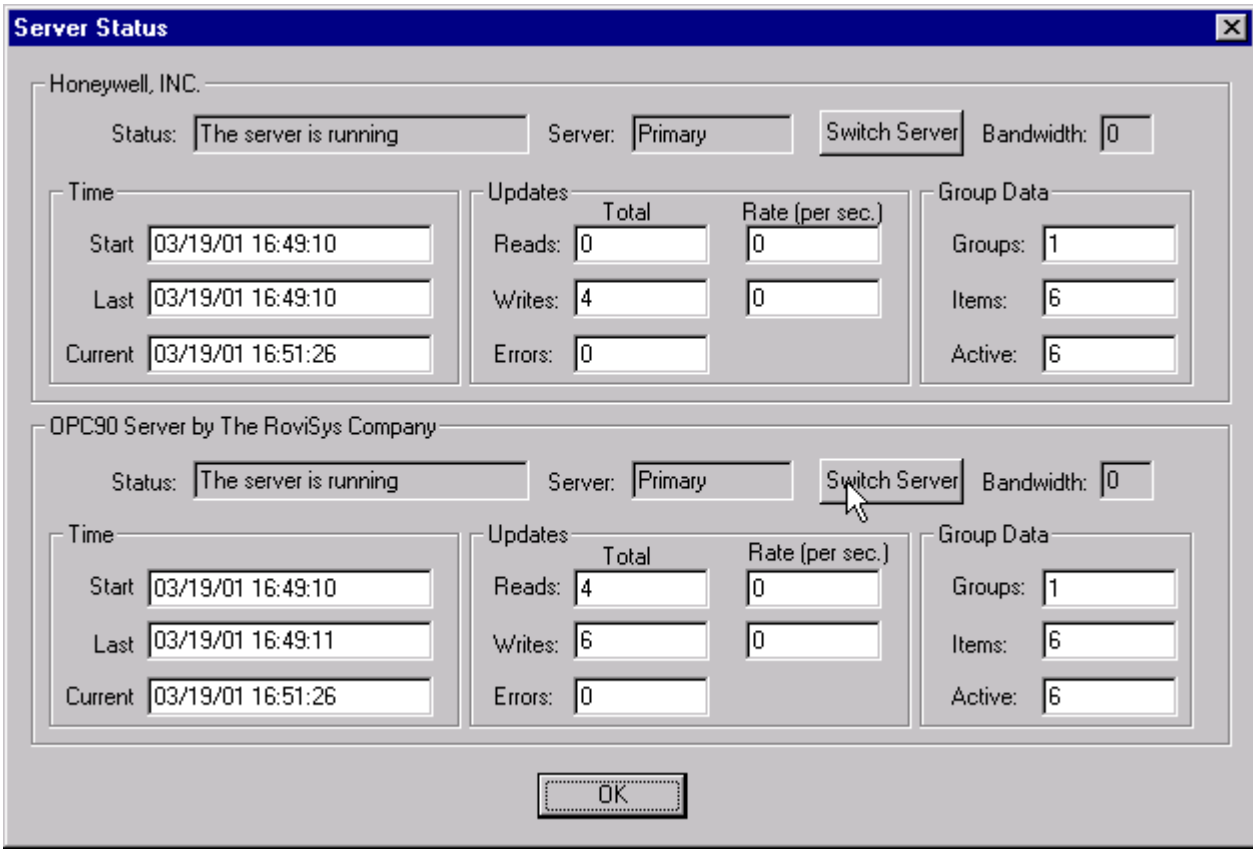


Figure 4.2 Server Status Dialog Box

A switchover to the redundant OPC Server can be forced by clicking **Switch Server**. If a redundant Server is not defined, ProcessLinx disconnects from the primary Server and then reconnects to it. If a redundant Server is defined, ProcessLinx attempts a connection to it. ProcessLinx attempts the connection for the "Number of Retries" specified in the **Utility ⇒ Configure** Options menu. After the number of retries has been exceeded, ProcessLinx then attempts to fail over to the first Server. This connection attempt / fail over process continues until a successful connection has been established. Messages for each failed attempt are logged to the ProcessLinx message pane and error log.

Verifying OPC Tags

In runtime, you can verify the status of all active OPC Items. Select **Utility ⇒ Verify OPC Tags** from the main menu to check all OPC Items. If an error is found, a message is displayed in the Message pane. A successful verification displays these messages:

MSG:: Verifying OPC Tags in Server A...
MSG:: OPC Tags Verified!
MSG:: Verifying OPC Tags in Server B...
MSG:: OPC Tags Verified

When you select the 'Verify Tags...' option from the Main Menu in runtime mode, ProcessLinx verifies each Pipe. For each Pipe both Servers are checked to see if there is a connection to them, if so then each Active Group of each client is verified. If not then an error is logged.

Verifying a group means that for each tag in each group ProcessLinx checks that the tag is active and that is has a connection to its corresponding item in the Server. If it does not, then a message displays for that item in the Message pane. A message also displays for each Inactive Group.

Ping

The ping function verifies access to a remote node. To ping a remote node, select **Utility ⇒ Ping...** from the main menu. In the Ping dialog box, enter the host machine's name or IP Address and click OK. The node's status is displayed in the Message pane. A successful Ping displays the message "*Ping>Host <localhost> is alive!*" where *localhost* is the desired host machine's name. A failed Ping message also indicates that the host is unreachable or is not responding due to a timeout or a network connection problem.

OPCShutdown

In runtime, ProcessLinx supports the IOPCShutdown interface and the Shutdown Request functionality from OPC Servers. If supported by the OPC Server, when the Server needs to shutdown, it can use this interface to send a shutdown request to ProcessLinx. ProcessLinx will then disconnect all items, groups and interfaces connected with that Server and will wait the amount of time specified by the 'Retry Wait Period' before attempting a connection to the redundant node (if configured).

If redundancy is not defined then ProcessLinx will attempt to connect to the original Server. The Server must be designed to not start up again (or deny access) if it is truly 'shutdown'. ProcessLinx will enter an infinite retry loop where it will continue to try reconnecting with that Server at the 'Retry Wait Period' until connection is re-established.

Notes:

Import/Export

This chapter provides ProcessLinx TagMap import/export examples. Using the Import/Export functions, you can edit configurations via text based, comma-separated-value (CSV) files such as the ProcessLinx configuration which is stored in a binary file.

To:	Select:
export an open ProcessLinx TagMap configuration	**File ⇒ Export CSV…** Enter an export filename and click OK. Edit the export file using Note Pad, Excel, etc. Be sure to follow the format and convention described later in this section.
TIP: Use **File ⇒ Save As…** to back up the existing ProcessLinx TagMap Configuration before attempting a file import operation.	

A ProcessLinx configuration can be imported into a new or an existing file. Open or create a configuration.

To:	Select:
import a configuration	**File ⇒ Import CSV…** Use the dialog box to find and select the desired CSV file. Click Open to import the new data. If the import function fails, an error message is displayed in the Message pane.
ProcessLinx checks the Pipe names in the import file against the Pipe names in the open configuration. If duplicate Pipe names are found, ProcessLinx prompts you as show in Figure 5.1.	

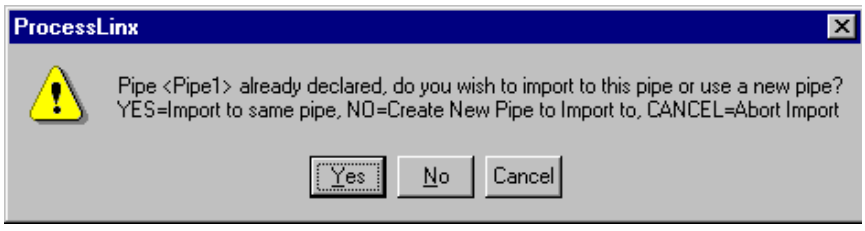

ProcessLinx
Pipe <Pipe1> already declared, do you wish to import to this pipe or use a new pipe?
YES=Import to same pipe, NO=Create New Pipe to Import to, CANCEL=Abort Import
Yes
No
Cancel

Figure 5.1 Import Confirmation

When you choose **Yes**, to overwrite the existing Pipe with the import information, ProcessLinx adds the new Groups, TagMaps, and Translation Tables to the existing Pipe. The following also occurs:

- All existing Groups and Translation Tables are overwritten with the settings from the CSV file.
- Existing TagMaps are not overwritten. ProcessLinx adds duplicate TagMaps.
- Delete TagMaps from the existing configuration before they are imported again.

If you choose **No**, to import the data to a new Pipe. ProcessLinx prompts you for a new Pipe name.

TIP

TagMaps can be dragged from one Group and copied or dropped into another Group.

Choose **Cancel**, to abort the import operation.

CSV File Formats

A CSV file is an ASCII Text file that contains fields delimited by commas. Text editors, spreadsheet programs, and database packages can create, edit, and save CSV files. Because commas are used as field delimiters, commas are NOT allowed in field values. Each text line represents one ProcessLinx command or object definition. A CSV file example with one Pipe, three Groups, and two Translation tables is given in Figure 4-2.

```
BEGIN_PIPE
PIPE,Pipe1,,1,localHost,HWHsc.OPCServer,localHost,HWHsc.OPC-
Server,5000,localHost,Rov iSys.OPC90Server,localHost,Rovi-
Sys.OPC90Server,5000
BEGIN_GROUPMAP
GROUPMAP,Outputs,,0,1,1.000000,1000,0,1.000000,1000,0
TAGMAP,,YHS103_B.DEVCTLA.MODE,,Device1.TIC108_B.MODE,,,0,0
TAGMAP,,HS_036.DEVCTLA.PV,,Device1.TIC108_B.PV,,,0,0
TAGMAP,,pidloop.pida.ophilm,,Device1.TIC108_R.HI_LIM,,,0,0
TAGMAP,,pidloop.pida.oplolm,,Device1.TIC108_R.LO_LIM,,,0,0
TAGMAP,,PIDLOOP.PIDA.PV,,Device1.TT108_R.SP,,,0,0
END_GROUPMAP
BEGIN_GROUPMAP
GROUPMAP,Inputs,,1,1,1.000000,1000,0,1.000000,1000,0
TAGMAP,,HS_036_EXTERNAL.DEVCTLA.PV,,Device1.FHS104_B.F1,,,1,0
TAGMAP,,LEVEL_EXTERNAL.NUMERIC18.PV,,Device1.WI113_B.OUT,,,1,0
TAGMAP,,PIDLOOP_EXTERNAL.NUMERIC6.PV,,Device1.TT108_B.OUT,,,1,0
END_GROUPMAP
BEGIN_GROUPMAP
GROUPMAP,READ-WRITE,,2,1,1.000000,1000,0,1.000000,1000,0
TAGMAP,,pidloop.pida.OP,,Device1.TIC108_R.OUT,,,0,1
TAGMAP,,pidloop.pida.MODE,,Device1.TIC108_R.MODE,,,0,0
TAGMAP,,pidloop.pida.SP,,Device1.TIC108_R.SP,,,0,0
TAGMAP,,pidloop.pida.PV,,Device1.TIC108_R.PV,,,0,0
TAGMAP,,HS_039.DEVCTLA.OP,,Device1.FHS107_R.OUT,,,0,1
TAGMAP,,HOPPER.DEVCTLA.OP,,Device1.YI109_R.OUT,,,0,1
TAGMAP,,PIDLOOP_EXTERNAL.PIDA.SP,,Device1.TIC108_B.SP,,,1,1
TAGMAP,,MIXER.DEVCTLA.OP,,Device1.YHS111_R.OUT,,,0,1
TAGMAP,,SPEED.DEVCTLA.OP,,Device1.YI130_R.OUT,,,0,1
END_GROUPMAP
END_PIPE
BEGIN_TABLE
TABLE,Device Output,DISCRETE
TABLE_VAL,5,1
TABLE_VAL,4,0
END_TABLE
BEGIN_TABLE
TABLE,Device Mode,DISCRETE
TABLE_VAL,1,1
TABLE_VAL,2,0
END_TABLE
```

Figure 5.2 Sample ProcessLinx CSV file

Pipe Declarations

Each Pipe object in the CSV file starts with BEGIN_PIPE and ends with END_PIPE. The Pipe is declared with a PIPE statement. The fields in the PIPE statement are in Table 5.A.

- A comma within the CSV file must separate each field.
- No string fields can contain commas.
- All string data types are NOT case sensitive.

Table 5.A Pipe Declaration Syntax

Field #	FIELD NAME	FIELD TYPE	FIELD DESCRIPTION	Examples
1	Line Type	String	Contains the string PIPE to define this line in the CSV file as a PIPE declaration.	PIPE
2	Pipe Name	String	Contains the unique name of the Pipe being declared. This field cannot contain any commas and is case insensitive.	Pipe1, Pipe2, Interface
3	Description	String	Specifies the description of this Pipe. This field cannot contain any commas. May be left blank.	User specified.
4	Enabled Flag	Integer	Pipe Enable flag, 1-Enable, 0-Disabled.	0 or 1
5	ProcessLogix Primary Node	String	Specifies the Host Name or IP Address of the Primary Node for the ProcessLogix Server.	LocalHost
6	ProcessLogix Primary Program ID	String	Specifies the OPC Program ID of the Primary Host for the ProcessLogix Server..	HWHsc.OPCServer
7	ProcessLogix Secondary Node	String	Specifies the Host Name or IP Address of the Secondary Node for the ProcessLogix Server.	167.34.56.77
8	ProcessLogix Secondary Program ID	String	Specifies the OPC Program ID of the Secondary Host for the ProcessLogix Server.	HWHsc.OPCServer
9	ProcessLogix Status Validation	Integer	Specifies the Status Validation period for the ProcessLogix Server (in milliseconds).	1000
10	Ext. Server Primary Node	String	Specifies the Host Name or IP Address of the Primary Node for the External Server.	LocalHost
11	Ext. Server Primary Program ID	String	Specifies the OPC Program ID of the Primary Host for the External Server.	RoviSys.OPC90Serve r
12	Ext. Server Secondary Node	String	Specifies the Host Name or IP Address of the Secondary Node for the External Server.	167.34.56.77
13	Ext. Server Secondary Program ID	String	Specifies the OPC Program ID of the Secondary Host for the External Server.	RoviSys.OPC90Serve r
14	Ext. Server Status Val.	Integer	Specifies the Status Validation period for the External Server (in milliseconds).	1000

Group Declarations

All Groups in a Pipe are declared between the BEGIN_PIPE and END_PIPE statements. A Group and its TagMaps are defined between the BEGIN_GROUPMAP and END_GROUPMAP statements. A Group is declared using the GROUPMAP line. Three Groups are defined in the example file shown in Figure 4-2. The fields in the GROUPMAP declaration are in Table 5.B.**T**

Table 5.B Group declaration syntax

Field #	FIELD NAME	FIELD TYPE	FIELD DESCRIPTION	Examples
1	Line Type	String	Contains the string GROUPMAP to define this line in the CSV file as a Group declaration.	GROUPMAP
2	Group Name	String	Contains the user-defined name of the Group being declared. This filed cannot contain any commas and is case insensitive.	Group1
3	Description	String	Specifies the description of this Group. This filed cannot contain any commas. May be left blank.	User specified.
4	Data Flow Direction	Integer	Integer value representing the data flow direction for this Group: 0 - () Data Flow from ProcessLogix to External Server 1 - (♉) Data Flow from External Server to ProcessLogix 2 - (♉) Data Flow Bi-Directional	0 or 1 or 2
5	Active Flag	Integer	Group Active flag, 1-Active, 0-Disabled.	0 or 1
6	ProcessLogix Deadband	Float	Specifies the Deadband (%) for the ProcessLogix Server side Group.	1.0000
7	ProcessLogix Update Rate	Integer	Specifies the Update Rate (in milliseconds) for the ProcessLogix Server side Group.	1000
8	ProcessLogix Time Bias	Integer	Specifies the Time Bias for the ProcessLogix Server side Group.	0
9	Ext. Server Deadband	Float	Specifies the Deadband (%) for the External Server side Group.	1.00
10	Ext. Server Update Rate	Integer	Specifies the Update Rate (in milliseconds) for the External Server side Group.	1000
11	Ext. Server Time Bias	Integer	Specifies the Time Bias for the External Server side Group.	0

TagMap Declarations

An unlimited number of TagMaps can be defined in each Group. Each TagMap is declared with a TAGMAP statement. The fields in a TAGMAP declaration are in Table 5.C.

Table 5.C TagMap Declarations

Field #	FIELD NAME	FIELD TYPE	FIELD DESCRIPTION	Examples
1	Line Type	String	Contains the string TAGMAP to define this line in the CSV file as a TagMap.	TAGMAP
2	Description	String	Specifies the description of this TagMap. This field cannot contain any commas. May be left blank.	User specified.
3	ProcessLogix OPC Item name	String	Specifies the OPC Item name to connect with in ProcessLogix.	YHS103_B.DEVCTL A.MODE
4	ProcessLogix Access Path	String	Specifies the ProcessLogix OPC Access Path to connect with. May be left blank.	
5	Ext. Server OPC Item Name	String	Specifies the OPC Item name to connect with in the External Server.	Device1.TIC108_B. MODE
6	Ext. Server Access Path	String	Specifies the Access Path to connect with in the External Server. May be left blank.	
7	Translation Table Name	String	Specifies the Translation Table name used for scaling by this TagMap. May be left blank if no scaling is required.	Device Output
8	Data Master	Integer	Specifies which server is the Data master if the Group's Data direction is set to bi-directional. Note, this setting is overwritten by the Groups data flow direction if not bi-direction. 0-ProcessLinx 1-External Server	0 or 1
9	Use Mode Attribute flag	Integer	Specifies that the TagMap will use the Mode Attribute OPC Item for sending data to ProcessLogix. 1-True, 0-False	0 or 1

Translation Table declarations

Translation tables are defined at the end of the CSV file. Each table is created with BEGIN_TABLE and END_TABLE declarations. The table name and type are declared with a TABLE line. The fields in a TABLE declaration are in Table 5.D.

Table 5.D Translation Table Declarations

Field #	FIELD NAME	FIELD TYPE	FIELD DESCRIPTION	Examples
1	Line Type	String	Contains the string TABLE to define this line in the CSV file as a Translation Table.	TABLE
2	Table Name	String	Specifies the unique table Name of this configuration. This field cannot contain any commas.	User specified.
3	Table Type	String	Specifies the Table Type. The only table type currently supported is DISCRETE.	DISCRETE

A TABLE_VAL statement declares each translation set in a table. The format for this line type is specified in Table 5.E.

Table 5.E Table_val syntax

Field #	FIELD NAME	FIELD TYPE	FIELD DESCRIPTION	Examples
1	Line Type	String	Contains the string TABLE_VAL to define this line in the CSV file as a Translation Value Map.	TABLE_VAL
2	ProcessLogix Discrete Value	Long Integer	Specifies the discrete value on the ProcessLogix Server side of the translation.	0
3	External Server Discrete Value	Long Integer	Specifies the discrete value on the External Server side of the translation.	1

Since you can not browse the OPC server address space within the ProcessLogix OPC Server (HWHsc.OPCServer); other methods have been designed to implement this function.

Within the Control Builder program, you must export the current configuration (or any desired tree level) that you need.

To export the current configuration, do the following:

1. Open Control Builder.
2. Select *File* ⇒ *Export* from the main menu to export the configuration or the desired branch(es).

3. In the TagMap dialog box in ProcessLinx click the Read from Export Directory to load in the browseable points.

Not all points will be loaded. This operation will only display points within the Control Builder configuration. Points (such as interface points) added using Quick Builder will not be displayed.

These points must be manually added on a single TagMap basis or can be added using the ProcessLogix Import function. All items within the TagMap dialog box displayed as nubs are connected as OPC Items.

Error Diagnosis and Debugging

This chapter explains how to diagnosis errors and debug ProcessLinx in the event the Runtime operations are not functioning properly. ProcessLinx logs errors along with informational messages to the message log in runtime mode only. Use the log to diagnose potential error conditions.

Quick Setup

Quick Setup describes how you can quickly setup ProcessLinx and diagnose common setup problems. The following checklist should be consulted if startup issues arise:

Table 6.A

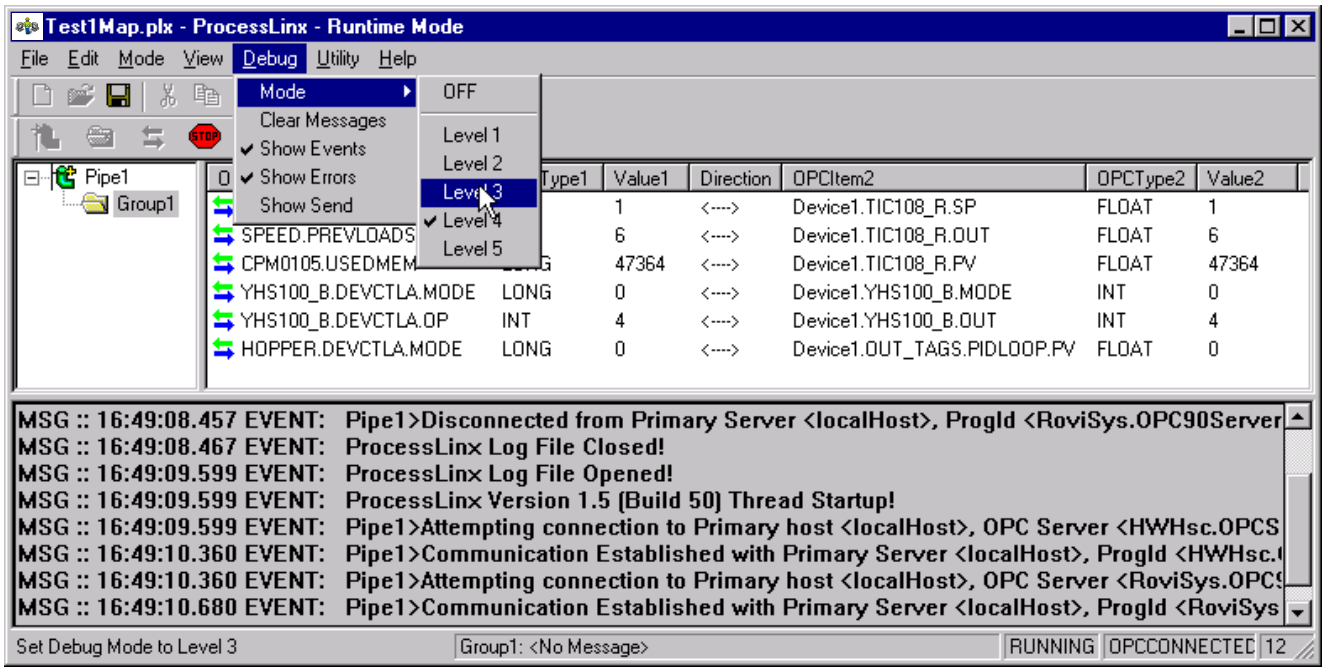
Item	Subject	Description
1.	Is TCP/IP enabled?	Verify TCP/IP protocol is enabled on the node running ProcessLinx and all OPC Servers.
2.	Are your OPC Servers setup properly?	Setup/Install both OPC Servers according to OEM specifications.
3.	Have you tried to ping a remote host?	• Attempt to ping the node running the desired OPC server you wish to connect with. • If you cannot ping a remote host then a network setup/hardware issue exists.
4.	Are your user accounts and security properly setup?	• If the network is a Domain, then the user account under which ProcessLinx is running needs to have the proper security setup in order to connect with the OPC Server(s) on the local/remote machine. • This user account needs to exist (be created under) that domain and thus have permissions to log in to the domain if both nodes are running under that domain. Consult the System Administrator of the domain for setting up user accounts.
5.	Is your Network a Workgroup?	• If the network is a Workgroup, the easiest way to connect the PCs it to place both nodes in the same workgroup and then define the same username and password on both nodes and then logon as the same user on each node. • If ProcessLinx is running a node A and an OPC Server is running on node B and both are part of the same workgroup then both node A and node B need to have a User account (with the same password) setup with the proper security privilege (in DCOM) to access and run the OPC Server on node B. ProcessLinx must be run under that user account. • Getting DCOM to work over a Workgroup is a challenge without totally removing all security. A user can be defined on both PCs yet still not work since each PC assigns an SID to the created user account. They will still be different SIDs even though it is the same user name. This is where having a Domain to do your authentication makes life easier.
6.	Is the user account added to the list of users in DCOM?	• The user account running ProcessLinx must be added to the list of users in DCOM with access permissions and launch permissions for that OPC Server. • Consult the Using DCOM OPC Servers for setting up this OPC Server security privilege.
7.	Have you configured each OPC Server?	• Configure each OPC Server if configuration is required. • If the OPC90 Server is used, you will need to configure the desired blocks first within the OPC90 Server and then save the configuration. For ProcessLogix, this is not necessary.

Table 6.A

Item	Subject	Description
8.	Is ProcessLogix the main OPC Server?	• You may need to create a user account <ps_user> and/or <mngr> with administrator privledges on the node running ProcessLinx. • If you are running an NT Workgroup, you may need to logon to the ProcessLinx node as <ps_user> or <mngr> in order to connect to the remote ProcessLogix OPC Server node. This is not the case when running under a domain. Using DCOM, the ProcessLogix OPC Server should already be setup to run under the <mngr> account, which is the reason why any remote Workgroup node wishing to connect with it needs to have the <mngr> admin account defined as a user. This step provides for the authentication of the ProcessLogix node to make callbacks (advise) into the ProcessLinx node. This is required if ProcessLinx is to receive callbacks via a ConnectionPoint Interface which is normal OPC method for receiving data from OPC Servers.
9.	Is ProcessLinx configured?	• Configure your ProcessLinx configuration. • Add a Pipe, one or more groups, add the desired tag maps for each group.
10.	Is ProcessLinx in runtime Mode?	• Place ProcessLinx in runtime mode.
11.	Has the Server started?	• If ProcessLogix OPC Server is used and all values are displayed as Bad Quality, then verify the ProcessLogix Server is 'Started' and running on the ProcessLogix node. • You may need to end the HWHsc.OPCServer (hscopcserv.exe) process on the host node first before going to run mode, assuming it was left in a hung state and no other clients are connected to it.
12.	Current connection does not support Advise!	• If a message appears in message log pane stating that 'Current connection does not support Advise! Defaulting to polling of OPC Server', then the problem may be due to a security issue as stated in #4, 5, 6 or 10 above.
13.	Does not support OPC Version 2.0 Interfaces.	• Upon startup, if ProcessLinx displays the message 'Does not support OPC Version 2.0 Interfaces' and the OPC Server is known to support v2.0 then verify #4, 5, 6 and/or 10 above and/or verify that the server is setup correctly on the server machine • Verify that any required client software is installed on the ProcessLinx node. • In most cases this is either the result of not having the proper security setup in DCOM and/or having older or missing software components in the system of the server.
14.	In Runtime, Server Status dialog states Total Reads=0.	• Using the Server Status dialog, in runtime if the Total Reads=0, then it is possible that the Advise subscriptions are not working correctly. • If problem cannot be corrected with one of above steps, then the Pipe and OPC Server may need to be setup for 'Polling'. • Keep in mind that advise connections will require security on the ProcessLinx node to be setup such that the user account which runs the OPC Server on the remote node will have 'permission' to send back data to the ProcessLinx node and thus to ProcessLinx. • This is not the same as the security setup to the Server in step #6. Since ProcessLinx is not an OPC Server, it cannot be setup with DCOM to enable such permission. • The network workgroup/domain will enable/define this privilege and thus that is why it is essential to correctly setup steps #4 or 5.
15.	ProcessLinx may fail to connect to the OPC Server because....	• ProcessLogix Server is stopped on the remote node. • Local machine had Norton AntiVirus on it, but it was disabled. The network used the POPProxy.dll provided by the AntiVirus software vendor to send/receive data and since it was disabled, then you could not ping the remote node. • ProcessLogix machine required a reboot. • Network cables were bad. • If 'Access is denied' messages appear in message log pane, then the most likely fix is solved in items 4, 5 and 6.

Debug Operations

You can execute simple debug operations during runtime from the main menu's debug menu item. The drop down list in the Main Menu below displays the following options:



In runtime mode, ProcessLinx supports the following debug options:

Table 6.B ProcessLinx Debug Options

Options	Description
Mode (Debug Level)	• Sets the debug mode of the ProcessLinx. • The higher the debug level the more messages will be printed to the screen. To turn all debug messages off set the MODE to OFF.
Clear Messages	• Clears all messages in the message pane window. • This should be done from time to time since the message pane window simply uses a buffer which keeps growing as new messages appear. The buffer size is limited only by the amount of memory available on the machine. It is not a circular buffer. • To clear all messages, click the toolbar. • If you wish to not worry about this then deselect the Show Events, the Show Errors and the Show Send and no messages will appear in this window.
Show Events	• Enables ProcessLinx to print out all event messages to the message log window (or file). • A check mark will appear next to this menu item when it is enabled. To disable this select it again so that the check mark disappears. • This option helps verify certain communication events from the ProcessLinx to the OPC Server(s).
Show Errors	• Enables ProcessLinx to print out all error messages to the message log window (or file). A check mark will appear next to this menu item when it is enabled. • If an error log directory was defined then leaving this option in the checked state may fill up that log file quickly. • This option helps verify communications with the OPC Server(s) and with the configuration.
Show Send	• Enables ProcessLinx to print out all writeBlock messages to the message log window (or file). If checked, then each write will be logged to the message pane. • This option helps verify certain communication events from the ProcessLinx to the OPC Server(s). • We do not recommend that you leave this option on too long. It may fill the message pane buffer and use significant memory resources. This could effect the performance of other tasks on the PC.

Using DCOM OPC Servers

This section details how to setup and diagnosis problems with DCOM Servers over a TCP/IP network. This is not an exhaustive guide and only discusses the common problems found with DCOM Network setup.

All of the software components of the ProcessLogix OPC Server are installed during the ProcessLogix server installation. The OPC Server option MUST be included with your ProcessLogix license. In order for ProcessLinx to connect to the ProcessLogix OPC Server over the network, the *OPC Server Connect* option must be installed on the client PC. Follow the directions outlined in the ProcessLogix Server and Client Installation Guide to setup a network connection correctly. If problems persist, this section may be used to connect your DCOM OPC Servers over the network.

In most cases, the TCP/IP network protocol must be installed and enabled on the PC before DCOM can be used. Before connecting to each OPC Server, the server must be in the Registry on the PC running ProcessLinx. The OPC Server generally performs this step at installation or when it is run for the first time.

If this does not occur, register the server:

1. Run regsvr32 <server.exe>" where server.exe is the name of the executable OPC Server.

2. The OPC Proxy DLL must also be registered on each PC using DCOM. Copy <opcproxy.dll> to the Windows System32 directory.

   In Windows NT, this directory is found at: C:\WINNT\System32. Run regsvr32 opcproxy.dll. A message displays stating that the registration succeeded. If the OPC Proxy is not registered on the local or remote node, a message displays indicating this at runtime startup or when a server switchover/fail-over occurs.

OPC Servers on Windows NT/Windows 2000

In Windows NT/Windows 2000, access to the OPC Server is setup using the <dcomcnfg.exe> (DCOM Configuration) program. In this program, the identity of the OPC Server must be set to Interactive User. This can be verified

under the Identity tab. Launching User is the default option, so this task must be done manually.

IMPORTANT This is not the case for the ProcessLogix OPC Server where the process is configured to run under the <mngr> account.

You must enable the security permissions correctly on the remote machine for the ProcessLinx node to access and launch the OPC Server remotely. Instructions for this procedure are specified in the *ProcessLogix Server and Client Installation Guide*.

1. Select the Security tab and verify that Use Custom Access Permissions is checked, and that the account being used to run the ProcessLinx node is selected as a User.

2. Verify that the same account is enabled under the Use Custom Launch Permissions checkbox.

3. If there is a problem adding a specific user in an NT Domain, the easiest way to enable these permissions (although not recommended) is to add the User Everybody for the two Groups mentioned above.

   However, this will allow any client/machine on the network to connect and launch the OPC Server remotely, leaving security wide open. If problems persist, consult your Network Administrator. The error E_ACCESSDENIED is displayed when running ProcessLinx if the required permissions are not enabled.

OPC Servers on Windows98 / Windows 95

In Windows 95, DCOM is disabled by default. Enable DCOM using 3 keys in the registry. The following is a description of these keys:

WARNING

Rockwell Automation holds no responsibility for damages to property and personnel or loss of time that may occur when making changes to the Registry.

- ***EnableRemoteConnect*** key on the OPC Server machine. Set to 'Y' for yes to allow ProcessLinx to connect to it remotely. This key is found from using the registry path:

  HKEY_LOCAL_MACHINE\Software\Microsoft\Ole\EnableRemoteConnect='Y'

- ***EnableDCOM*** key. Set to 'Y' on BOTH the remote OPC server machine and the machine running ProcessLinx regardless of the Windows version. This key is found using the registry path:

  HKEY_LOCAL_MACHINE\Software\Microsoft\Ole\EnableDCOM='Y"

- ***LegacyAuthenticationLevel*** key. Set to 1 on the remote OPC server machine. This key is found using the registry path:

  HKEY_LOCAL_MACHINE\Software\Microsoft\Ole\LegacyAuthenticationLevel=1

WARNING

For Windows 95 machines, this key setting allows access from any remote DCOM clients, so care must be taken to ensure the network is secure and/or a private intranet.

- Reboot the OPC server machine and the machine running ProcessLinx after making registry changes.

- On Windows 95 machines, RPCSS.EXE must be pre-launched in order for DCOM to work. This process may not be started automatically depending upon the Version of the OS. If this is the case, place RPCSS.EXE in a named value as an executable that will be started automatically at Windows 95 startup. To cause the executable to launch when the shell loads, place the named value of "RPCSS.EXE in the registry under the key:

  HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run

- To launch the executable immediately when booting the machine (before login), place the named value under the registry key:

KEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices

- On Windows 95 machines only, the external OPC Server MUST BE RUNNING in order for ProcessLinx to connect with it. ProcessLinx can launch or start an OPC Server remotely based on security permissions on a Windows NT or Windows 2000 machine.

Notes:

How Are We Doing?

Your comments on our technical publications will help us serve you better in the future. Thank you for taking the time to provide us feedback.

You can complete this form and mail it back to us, visit us online at www.ab.com/manuals, or email us at RADocumentComments@ra.rockwell.com

Pub. Title/Type ProcessLinx OPC Server to Server Software

Cat. No. 1757-PROCESSLINX Pub. No. 1757-UM002B-EN-P Pub. Date January 2003 Part No. 957726-17

Please complete the sections below. Where applicable, rank the feature (1=needs improvement, 2=satisfactory, and 3=outstanding).

Overall Usefulness	1	2	3	How can we make this publication more useful for you?
Completeness (all necessary information is provided)	1	2	3	Can we add more information to help you? procedure/step illustration feature example guideline other explanation definition
Technical Accuracy (all provided information is correct)	1	2	3	Can we be more accurate? text illustration
Clarity (all provided information is easy to understand)	1	2	3	How can we make things clearer?
Other Comments				You can add additional comments on the back of this form.

Your Name _____ Location/Phone _____

Your Title/Function _____ Would you like us to contact you regarding your comments?

___No, there is no need to contact me

___Yes, please call me

___Yes, please email me at _____

___Yes, please contact me via _____

Return this form to: Allen-Bradley Marketing Communications, 1 Allen-Bradley Dr., Mayfield Hts., OH 44124-9705

Phone: 440-646-3176 Fax: 440-646-3525 Email: RADocumentComments@ra.rockwell.com

PLEASE FASTEN HERE (DO NOT STAPLE)

Other Comments

PLEASE FOLD HERE

PLEASE REMOVE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 18235 CLEVELAND OH

POSTAGE WILL BE PAID BY THE ADDRESSEE

Allen-Bradley
RELIANCE ELECTRIC
DODGE
ROCKWELL SOFTWARE
Rockwell Automation

1 ALLEN-BRADLEY DR
MAYFIELD HEIGHTS OH 44124-9705

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
Europe/Middle East/Africa: Rockwell Automation, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication 1757-UM002B-EN-P - January 2003 PN 957726-17
Supersedes Publication 1757-UM002A-EN-P - May 2001
Copyright © 2003 Rockwell Automation, Inc. All rights reserved. Printed in the U.S.A.